



YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ
ELEKTRİK MÜHENDİSLİĞİ BÖLÜMÜ

Bitirme Tezi

EV OTOMASYON SİSTEMİ

Proje Danışmanı: Prof. Dr. Halit PASTACI

01012197 Serdar ATEŞ

İstanbul
2006©

İÇİNDEKİLER

ÖNSÖZ

ÖZET

ABSTRACT

BÖLÜM 1: GİRİŞ

BÖLÜM 2: PROJE BLOK DİYAGRAMI VE SİSTEMİN ÇALIŞMASI

BÖLÜM 3: PIC16F84A MİKRODENETLEYİCİSİ

BÖLÜM 4: RS-232 SERİ HABERLEŞMESİ

BÖLÜM 5: TELEFON HATTI İLE CİHAZ KONTROLÜ

5.1 TELEFON HATTI NASIL ÇALIŞIR?

5.1.1 Telefon Hattının Özellikleri

5.1.2 Neden Telefon Hattında 48 V Kullanılır?

5.1.3 Telefon Hattının Bant Genişliği Nedir?

5.2 ÇAĞRININ GELİŞİMİ

5.3 TELEFON HATLARINDA KORUMA

5.3.1 Varistör

5.3.2 Zener Diyod

5.3.3 Gas Arrester

5.4 TELEFON RING DEVRESİ

5.5 DTMF

5.5.1 DTMF Frekansları

5.6 ÖLÇÜM SONUÇLARI

5.7 MODÜLÜN ÇALIŞMA PRENSİBİ

5.8 KAYNAK KODLAR

BÖLÜM 6: GRAFİK KULLANICI ARAYÜZÜ

BÖLÜM 7: YAZILIMLAR

7.1 KONTROL YAZILIMI

7.2 GERİ BESLEME YAZILIMI

BÖLÜM 8: PROJE DONANIMI

SONUÇ

KAYNAKLAR

ÖZGEÇMİŞ

ÖNSÖZ

Bu bitirme çalışması, yaşamımın en önemli kısımlarından biri olan Elektrik Mühendisliği Lisans eğitiminin son noktasıdır. Mühendislik yaşamımda iyi bir referans olacağını düşündüğüm bu çalışma, lisans eğitimin boyunca almış olduğum temel mühendislik derslerinin yanı sıra daha önce gerçekleştirmiş olduğum ödev ve proje çalışmalarında kazandığım deneyimlerimi en üst düzeyde bir araya getirmem suretiyle gerçekleşmiştir.

Çalışmalarım sırasında maddi ve manevi anlamda hiçbir yardımı esirgemeyen tez danışmanım Prof. Dr. Halit PASTACI'ya, Emre SUAT'a, Haluk ECE'ye, Ali Uğur EROL'a, Eda ÇETİNKAYA'ya ve Elektrik Mühendisleri Odası İstanbul Şubesi'ne, Robotik ve Otomasyon Kulübü'ne ve sevgili aileme teşekkürü bir borç bilirim.

Bu bitirme tezi çalışması sırasında yalnızca teknik becerilerimi geliştirmekle kalmayıp, bir projenin mali ve diğer yönlerinin de nasıl yönetilip sonuçlandırılacağını ve büyük meselelerin küçük adımlar ile çözülebileceğini gördüm.

Serdar ATEŞ
İstanbul, Haziran 2006

ÖZET

Hazırlanan bu projede bir ev maketi üzerinde bilgisayar tabanlı, telefon hattı üzerinden uzaktan kontrol edilebilen bir ev otomasyon sistemi gerçekleştirilmiştir.

Projenin amacı günümüz otomasyon teknolojilerine katkı sağlayabilmektir. Bu proje tek bir kumanda merkezinden veya evden uzakta olmamıza rağmen evde gerçekleştireceğimiz otomasyon sistemini kumanda olanağı sağlamaktadır.

Proje kapsamında tasarlanan elektronik donanım, RS-232 seri haberleşme protokolünü kullanarak yazılmış grafik kullanıcı arayüzünün kurulu olduğu kişisel bilgisayar ile haberleşmekte ve ev maketi üzerinde modellenen cihazların elektriki kontrollerini gerçekleştirmektedir.

Projenin devre tasarımı gerçekleştirilirken RS-232 seri haberleşme protokolü hakkında bilgi edinildi. Telefon hattı ile haberleşme kısmı gerçekleştirilirken telefon hattının elektriki özellikleri çok iyi analiz edilmiştir. Projenin gerçekleştirildiği Yıldız Teknik Üniversitesi'nin iç santrali üzerinde elektriki ölçümler yapılmıştır.

Devre tasarımı yapılırken proje birkaç modül halinde düşünüldü. Bunlar modellenen elektrikli cihazların bilgisayar vasıtasıyla kontrolünü sağlayan modül, cihaz ve sensörlerden gelen geri besleme bilgisini bilgisayara aktaran modül ve telefon hattı vasıtasıyla kontrolü sağlayan uzaktan kontrol modülüdür.

Bilgisayardaki grafik kullanıcı arayüzü Microsoft Visual Basic 6.0 programı kullanılarak gerçekleştirilmiştir ve gerçek zamanlı olarak elektronik donanım ile haberleşmektedir.

ABSTRACT

In this project, a computer-based home automation system which is also controlled via phone line is demonstrated.

The aim of this project is to contribute to current automation technology. This project lets us control a home automation system from one command center despite the fact that we are not physically related with home.

According to the aim mentioned above, electronic side of this project provides us to communicate with the computer which has the graphical user interface software via RS-232 serial communication protocol and controls the devices presented with a home model.

While designing the electronic circuit, knowledge about RS-232 has been gained. While designing the phone line communication part, the electrical characteristics of the phone line has been analyzed carefully and measurements have been practiced in Yildiz Technical University where the project has been verified.

Electronic hardware consists of some parts. These parts are the first part which lets us control electrical devices via personal computer, and the second part which sends the feedback data coming from the sensors and the devices, and the third one is the remote control part that lets us control via phone line.

The graphical user interface software is programmed with Microsoft Visual Basic 6.0 and communicates with the electronic hardware in real time.

BÖLÜM 1: GİRİŞ

Hazırlanan bu projede günümüzün gözde teknolojilerinden biri olan akıllı evler hakkında bir çalışma gerçekleştirilmiştir. Bilindiği üzere ev otomasyon sistemleri günümüzde çeşitli katma değerler yaratan ve özellikle nüfusun yoğun olduğu büyük iş yerlerinde ve diğer komplekslerde yönetim ve denetim kolaylığı sağlamaktadır. Ancak böyle bir çalışma çok kapsamlı olacağından, gerek zaman gerekse de maliyet unsurları göz önünde bulundurulduğunda bu işlem daha farklı bir şekilde modellenmiştir.

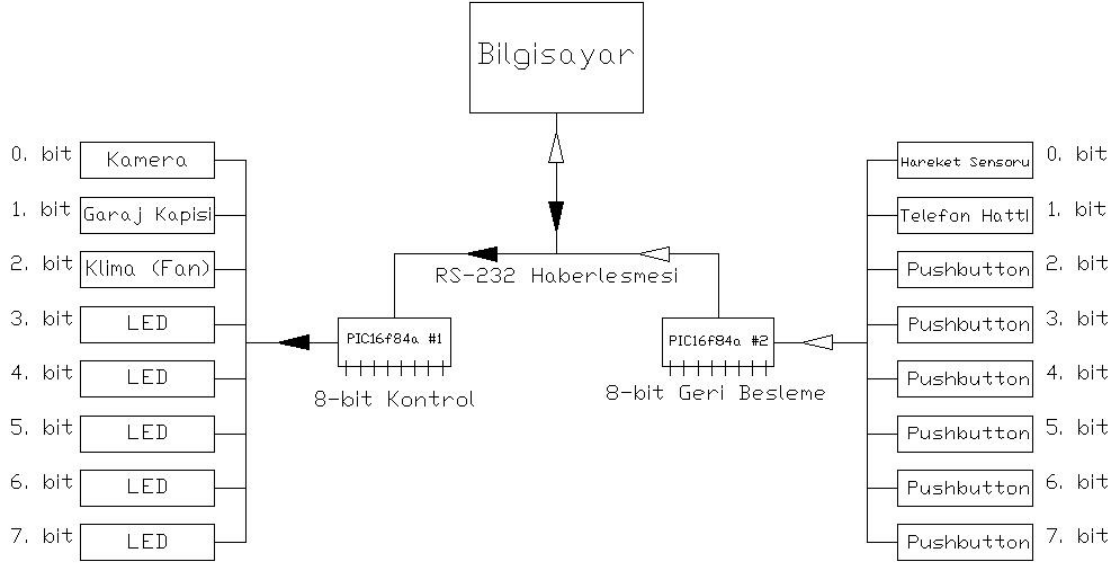
Bu projede bir ev maketi üzerinde RS-232 seri haberleşme protokolü kullanılarak Visual Basic 6.0 ile hazırlanmış bir grafik kullanıcı arayüzü ile haberleşen bir model oluşturulmuştur. Bu modelde 8-bit aç/kapa kontrolü ve 8-bit geri besleme gerçekleştirilmiştir. 8-bit aç/kapa kontrolünün yapıldığı mikrodenetleyicinin çıkışlarının bir kısmı ev otomasyonu uygulamalarına uygun bir şekilde kullanılmış, geriye kalan uçlara LED'ler bağlanmıştır. Bunun amacı proje maliyeti ile projenin görselliğinin sağlayacağı fayda dengesini sağlamaktır. Ancak bilindiği üzere bu tür uygulamalarda önemli olan çıkışlarda ne tür cihazların sürüldüğü değil, çıkışların kontrolünü istenildiği şekilde sağlayabilmektir.

Yine aynı şekilde ve yukarıda bahsedilen nedenlerden ötürü 8-bit geri beslemenin alındığı uçlardan bir kısmı kullanılmış, geri kalan kısımlarına pushbuttonlar bağlanmıştır.

Ancak proje hazırlanırken hiçbir donanımsal değişikliğe gerek kalmadan, yalnızca yazılım değişikliği yapılarak modüler bir sistem tasarlanması hedeflenmiştir.

BÖLÜM 2: PROJE BLOK DİYAGRAMI VE SİSTEMİN ÇALIŞMASI

Projenin temel yapısını gösteren olan blok diyagram Şekil 2.1'deki gibidir.



Şekil 2.1: Sistem Blok Diyagramı

Yukarıdaki şekilde görüldüğü gibi, sistem temel olarak 8-bit aç/kapa kontrolünü gerçekleştiren mikroişlemci, 8-bit geribeslemenin alındığı mikroişlemci, bu bilgilerin bir grafik kullanıcı arayüzü ile işlendiği bilgisayar ve bu modüllerin haberleşmesini sağlayan RS-232 seri haberleşmesinden meydana gelmektedir.

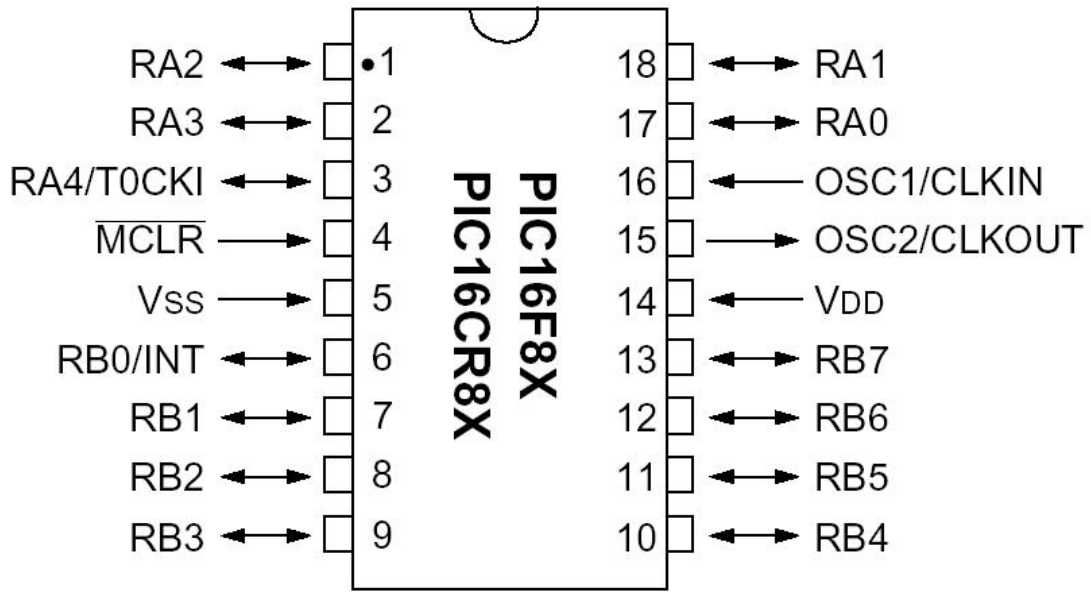
Bu işlemleri gerçekleştiren mikroişlemciler Microchip firmasının PIC16F84a mikroişlemcisidir. Bu mikroişlemciler hakkında bilgi ileriki bölümlerde verilecektir.

Sistemin Bilgisayar kısmı Microsoft Visual Basic 6.0 üzerinde hazırlanmış görsel bir programdan oluşur. Bu programda istenilen çıkışlar değiştirilmekte, sisteme gelen geribesleme bilgisi işlenmektedir.

Bilgisayar ile sistemin bağlantısı ileriki bölümlerde açıklanan MAX-232 ile gerçekleşir. Bu entegrenin amacı TTL seviyesindeki lojik sinyalin, seri haberleşme seviyesindeki gerilimlere çevrilmesini sağlamaktır.

BÖLÜM 3: PIC16F84A MİKRODENETLEYİCİSİ

PIC16F84A Mikrodenetleyicisinin bacak bağlantıları Şekil 3.1’de görüldüğü gibidir.



Şekil 3.1: PIC16F84A Mikrodenetleyicisinin Bacak Bağlantıları

Microchip firması tarafından üretilen PIC16F84A mikrodenetleyicisi en popüler ve en yaygın olarak kullanılan PIC üyesidir. Bacak tanımları Tablo 3.1’de gösterildiği gibidir.

Bacak İsmi	Bacak No	Tanım
RA0	17	PORTA bit 0
RA1	18	PORTA bit 1
RA2	1	PORTA bit 2
RA3	2	PORTA bit 3

RA4/TOCKI	3	PORTA bit 4, TMR0 zamanlayıcısı saat girişi
RB0/INT	6	PORTB bit 0, INT dış kesme bacağı
RB1	7	PORTB bit 1
RB2	8	PORTB bit 2
RB3	9	PORTB bit 3
RB4	10	PORTB bit 4
RB5	11	PORTB bit 5
RB6	12	PORTB bit 6
RB7	13	PORTB bit 7
VSS	5	Toprak Bacağı
VDD	14	Pozitif Kaynak Bacağı
MCLR	4	Reset Bacağı
OSC1	16	Osilatör Giriş Bacağı
OSC2	15	Osilatör Giriş Bacağı

Tablo 3.1: PIC16F84A Mikrodenetleyicisinin Bacak Tanımları

PIC16F84A Mikrodenetleyicisinin şu özellikleri vardır:

- 1 KB Flash program bellek
- 68 byte RAM bellek
- 64 byte EEPROM bellek
- 14 bit genişliğinde komutlar
- Kesme (Interrupt) Kaynağı
- 13 giriş/çıkış portları
- 25 mA port çıkış akımı
- Bekçi Köpek (Watchdog timer) Devresi
- Uyku Modu
- +5 V'da 2 mA akım, 2 V'da 15 nA akım

PIC16F84A mikrodenetleyicisi 10 MHz frekansa kadar bir saat hızında çalışabilir. Bu hızda çalışınca komut çevrimi (machine cycle) 400 ns kadardır. RISC yapısına sahip olan bu mikrodenetleyicinin sadece 35 tane tek kelimelik komutu vardır. PIC16F84A mikrodenetleyicinin 4 tane kesme (interrupt) kaynağı bulunur. Bunlar bu şekilde olabilirler:

- Dıştan RB0/INT bacağı ile
- TMR0 zamanlayıcısının taşması ile
- PORTB 4-7 bacaklarında olan herhangi bir değişiklikten dolayı
- EEPROM yazma işleminin tamamlanması ile

PIC16F84A mikrodenetleyicisinin toplam 13 tane portu vardır. Bunlardan 5 tanesi PORTA'ya, 8 tanesi de PORTB'ye aittir.

Projede temel olarak 8 kontrol biti, 8 de geri besleme biti kullanılmıştır. Bu yüzden 16 port gerekmektedir. Bunun için bacaklardan birine port çoklayıcı konulabilirdi ya da bacak sayısı daha fazla olan bir mikrodenetleyici kullanılabilirdi. 2 adet PIC16F84A kullanılmasının sebebi maliyetinin son derece düşük olması (tanesi yaklaşık 3 YTL) ve tek bir mikrodenetleyici kullanıldığında program kodlarının çok daha karmaşık hale gelmesiydi. Böylelikle proje maliyeti hem minimum seviyede tutulmuş, hem de mikrodenetleyici kodları kayda değer miktarda kısaltılmıştır.

BÖLÜM 4: RS-232 SERİ HABERLEŞMESİ

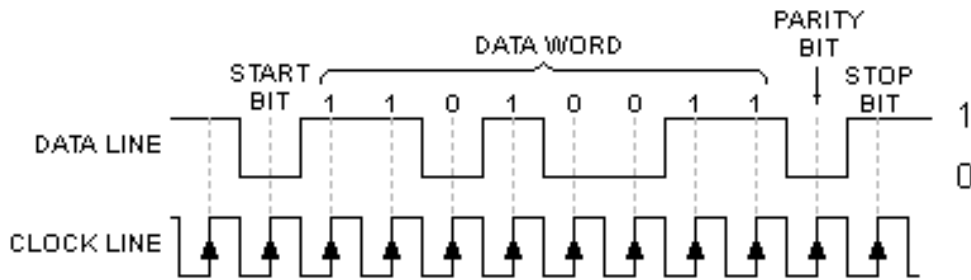
RS-232 Haberleşmesi tek bir amaç için üretilmişti; DTE (Data Terminal Equipment) ve DCE (Data Communications Equipment) arasında iletişim sağlamak. Burada DTE bilgisayarı temsil etmekte, DCE ise ona bu yoldan bağlanacak olan modem, mouse vb. cihazları tanımlamaktadır.

RS-232 nin en büyük özelliği verilerin bir seri bir hat üzerinden iletilmesidir. İki türlü seri haberleşmeden bahsedilebilir. Bunlardan biri Eşzamanlı haberleşme (synchronous

communication), Diğeri ise Eşzamanlı olmayan Haberleşme (asynchronous communication) . Eş zamanlı haberleşmede veri sinyaline eş zamanlı olarak bir clock sinyali mevcuttur. Böylece verinin gönderildiği taraf, bir yandan kendi clock sinyalini göndererek senkronizasyonu sağlar. Fakat günümüzde daha çok kullanım alanı bulan seri haberleşme tipi Eşzamanlı olmayanıdır. Buradaki fark şudur, iki taraf arasında bir clock sinyali alış verişi olmayacaktır. Buna karşın iki tarafta aynı hızda (rate) veri iletimi yapacaktır, clock sinyali aslında hayali olarak mevcuttur, ama iletilmez. Bunun dışında veri paketinin başına Start Biti olarak tanımlanan bir bit uzunluğu ile aynı veya daha uzun bir bit eklenecek, aynı şekilde Stop Biti olarak sona bir diğer bit eklenecek, eğer gerek görülürse veri kontrolünü sağlamak üzere, Stop biti ile verinin son biti arasına Parity biti olarak tanımlanan kontrol biti konabilir.

Parity biti 5 farklı şekilde tanımlanabilir. Sürekli logic “1” değerini taşıması istenebilir (mark parity bit), sürekli logic “0” değeri taşıyacak şekilde olabilir (space parity bit), bir diğer kullanım şekli, veri bitlerinin toplamı çift ise logic “1” değerini taşımasıdır. (even parity bit), veya veri bitlerinin toplamı tek ise logic “1” değerini taşıması istenebilir (odd parity bit), bunların dışında parity biti bulunmayabilir (non parity bit frame).

Bütün bu anlatılanları temsilen Şekil 4.1 gösterilebilir.

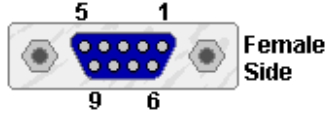


Şekil 4.1 Veri Akış Şekli

Şekil 4.1 de görülen clock line mevcut değildir. Veriler bu clock line a senkronize olacak şekilde ayarlanır. Bu haberleşmede hızdan bir saniye içinde gönderilen bit miktarı anlamına gelen Bps (Bits Per Second) birimi ile söz edilir. Bu hızlar belli bir standartta

gitmektedir. 600, 1200, 2400, 4800, 9600, 19200 ve 38400 gibi. Bu projede hıza gereksinim bulunmadığından, genelde çok kullanılan 9600 bps tercih edilmiştir. Parity biti de kapatılmıştır (non parity bit frame).

Bilgisayarda kullanılan soket tiplerinden bahsetmek gerekirse bugün hala kullanılan DB-9 ile kullanılabilirliğini yitiren DB-25 soketleri mevcuttur. Bu projede DB-9 tipi soket kullanılmaktadır. Bu soketteki pinlerin dağılımı aşağıdaki gibidir.



- 1-Carrier Detect
- 2-Received Data
- 3-Transmitted Data
- 4-Data Terminal Ready
- 5-Signal Ground
- 6-Data Set Ready
- 7-Request To Send
- 8-Clear To Send
- 9-Ring Detector

Şekil 4.2 DB9 Dişi Soketinin Ayak Dağılımı

Daha açıklayıcı olması açısından DTE nin bilgisayar olduğunu düşünülürse, DCE de bağlanacak olan cihaz ise, Tablo 4.1 deki semboller kullanılarak pinlerin amacı açıklanabilir.

 ← Bilgisayara gelen veri	 → Cihaza giden veri
--	---

Tablo 4.1

Carrier Detect  ←

Bu pin ile Cihaz bilgisayara kendi tarafındaki bağlantı olayının tamamlandığını anlatır (Aslında Harici bir modem düşünülürse, bu durum daha manalı olacaktır. Modem internete veya başka bir sisteme bağlandığını bilgisayara bu pin ile bildirir.)

Received Data 

Bu pin verilerin cihazdan bilgisayara gönderildiği pindir.

Transmitted Data 

Bu pin verilerin bilgisayardan cihaza gönderildiği pindir.

Data Terminal Ready 


Bu pin ile bilgisayar cihaza haberleşme için hazır olduğunu bildirir. Eğer bilgisayar herhangi bir sebeple bu hattın logic "0" değeri gönderirse cihaz haberleşmeyi durdurur.

Signal Ground

Bu pin iki tarafın toprak seviyesini eşit hale getirmektedir.

Data Set Ready 

Cihaz bu pin ile bilgisayara data göndermeye hazır olduğunu söyler.

Request To Send 

Bilgisayar veri alma isteğini cihaza bu pin ile bildirir.

Clear To Send 

Cihaz bu pin ile bilgisayardan veri almaya hazır olduğunu söyler.

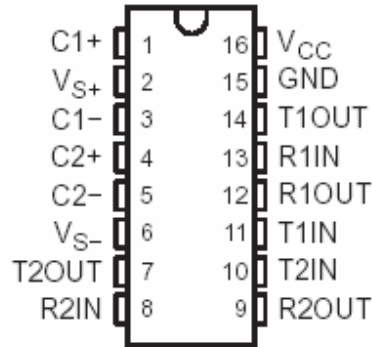
Ring Detector  ←

Tamamıyla modem bağlantısı için hazırlanmıştır, bu pin ile bağlı olan modem hattında telefonun çaldığı bildirilir.

Adından da anlaşılacağı gibi Received Data ve Transmitted Data pinleri ve de bunların geri dönüşü olan Signal Ground burada en önemli rolü oynamaktadır. RS-232 ilk olarak bilgisayarlara modem bağlantısını sağlamak için üretildiğinden, diğer bütün pinler bir modemle sağlanacak bağlantı için hazırlanmıştır. Fakat sistemin düzgün çalışabilmesi için, bağlanacak cihazda (DCE) 1,4,6 numaralı pinleri ve de 7,8 i kendi aralarında bağlayarak, DTE (yani bilgisayar) aldatılmaktadır. (bir modem bağlanmış izlenimi yaratılmaktadır.) Bu işlem yapıldığında haberleşme için toplam 3 pin kullanılmış olmaktadır.

Bunun dışında haberleşmede çıkış seviyeleri logic “1” için $-5V$ ile $-15 V$ arası bir değerle temsil edilir. Logic “0” değeri ise $+5V$ ile $+15 V$ arası bir değer ile ifade edilir. Kabul edilen giriş seviyeleri ise, logic “1” için $-3V$ ile $-15V$ arası, logic “0” için $+3V$ ile $+ 15 V$ arasındadır. Bu sayede haberleşme çok uzun mesafeler arasında çalışabilmektedir.

Bu logic seviyeler TTL girişli ve çıkışlı sistemler olan çoğu mikro denetleyici ile çalışmayacağı gibi zarar da verecektir. Bu yüzden çok yaygın olarak kullanılan MAX-232 entegresi bu projede kullanılmaktadır. Bu entegrenin bacak bağlantıları Şekil 4.3 te gösterilmiştir.



Şekil 4.3 MAX232 Entegresinin Bacak Bağlantıları

Bu entegrenin amacı, TTL seviyesindeki sinyalleri RS-232 haberleşmesinde istenen gerilim seviyesine getirmektir (Lojik “1” için –3 ile –15 V arası, Lojik “0” için +3 ile +15 V arası). Ayrıca tersleme işlemi de yapılmaktadır.

Bu entegrenin amacı, öncelikle RS-232 haberleşmesi için gerekli olan negatif ve pozitif gerilim seviyelerini elde etmektir. Besleme gerilimi olarak +5V kullanılmaktadır. Katalog değerlerinde bu gerilimler tipik olarak +7/-7 V olarak verilmiştir. Bu işlemi yapacak olan devrenin kapasiteleri (1µF olarak verilmiştir), kılıf içine yerleştirilemediği için, dışarıdan bağlanması için bacaklar ayarlanmıştır.

BÖLÜM 5: TELEFON HATTI İLE CİHAZ KONTROLÜ

Ev otomasyon sistemlerinden beklenen özelliklerden biri de uzaktan denetim gerçekleştirilebilmesidir. Bu da denetimin yapılacağı mesafe göz önüne alındığında birkaç şekilde yapılabilir. Bunlar RF ile kontrol, telefon hattı üzerinden kontrol, internet tabanlı kontrol, ve benzeridir. Bu çalışma için gerek maliyetinin düşük olması gerek de uygulamasının kolay olması açısından telefon hattı üzerinden kontrol tercih edilmiştir.

Bu bölümde anlatılan kısımlar tez yazarının Proje – II çalışması olup, daha önce başarıyla gerçekleştirilmiş bir çalışmadır. Bu yüzden bu bitirme çalışmasına bir modül olarak eklenmesi hem projenin esnekliğini arttırması açısından hem daha gerçek bir model oluşturulması açısından uygun görülmüştür. Ayrıca bu modülün donanım ve yazılımı bu BÖLÜM-5 başlığı altında incelenerek, tez çalışmasının karmaşık bir hale gelmesini önlenmiştir.

Bu modülün devre tasarımını gerçekleştirebilmek için telefon hattının elektriki özellikleri çok iyi analiz edilmiştir. Projenin gerçekleştirildiği Yıldız Teknik Üniversitesi'nin iç santrali üzerinde elektriki ölçümler yapılmıştır.

Devre tasarımı yapılırken bu modül birkaç altmodül halinde düşünöldü. Bunlar hatta çağrı gelip gelmediğini kontrol eden, çağrı geliyorsa hattı açan, telefon hattı üzerinden gelen DTMF sinyallerini sayısal olarak kodlayan ve bu süreci kontrol eden mikrokontrolör birimleridir.

Bu modül hakkında detaylara girmeden önce bilinmesi gereken bazı temel bilgiler vardır. Bunlar, telefonda veri iletiminin temel prensibi, telefon hattının özellikleri, DTMF sinyalleri, bu sinyallerin üretilmesi, kodlanması ve çözölmesidir.

5.1 TELEFON NASIL ÇALIŞIR?

Telefon, evinizden bir başka arkadaşınıza elektrik akımı ile ses taşıyan cihazdır. Bir arkadaşınızla telefonda konuşurken, telefon şirketi arkadaşınızın hattı ile sizin hattınız arasında sabit bir akım dolaştırır. İki telefon, sizin ve arkadaşınızın telefonları, aynı sabit akımı kullanırlar. Fakat telefonun mikrofonuna konuştuğunuzda, sizin telefonunuz ile telefon şirketi arasında akan akım yükselip, alçalır. Bu yükselip alçalmalar, doğrudan sesinizin telefonunun mikrofonunda yarattığı titreşimlerle ilgilidir.

5.1.1 Telefon Hattının Özellikleri

Telefon hattı TIP ve RING olarak adlandırılmış iki telden oluşur. Bu iki tel üzerinde iki farklı DC seviyesi mevcuttur. Bu sayede hattın dengeli bir hat özelliğine kavuşması ve şaselemeden dolayı oluşabilecek gürültü ve zayıflamaların en aza indirgenmesi sağlanmıştır.

Hattın AC empedansı 600 Ω kadardır. Telefon ON-HOOK konumunda hatta -48 V'luk DC gerilim mevcuttur ve hattan akım akmaz. Telefon OFF-HOOK yapıldığında hat voltajı telefonun 100-200 Ω civarında olan DC empedansı üzerinde akan 20-80 mA akım dolayısı ile 6-10 V DC seviyeye düşer.

5.1.2 Neden Telefon Hattında 48 V Kullanılır?

Kilometrelerce öteye ince bir telefon telinden güvenli bir şekilde veri aktarabilmek için - 48 V yeterli olduğu için seçilmiştir. Birçok ülkede, düşük gerilimle çalışan devrelerin güvenli olması için, 50 V (DC) seviyesinin altı seçilir. Ayrıca 48 V'u üretmek çok kolaydır (4 tane 12 V'luk araba aküsü seri bağlanarak üretilebilir).

Fakat ülkeden ülkeye bu gerilim seviyesinde değişiklikler olabilir. 36 V ile 60 V arasındaki gerilimler değişik ülkelerde kullanılmaktadır.

5.1.3 Telefon Hattının Bant Genişliği Nedir?

Telefon hattının bant genişliği Avrupa ülkeleri ile ABD'de 3 kHz'dir. Telefon hattı vasıtasıyla 400 Hz ile 3.4 kHz frekanslı sinyaller iletilebilir.

5.2 ÇAĞRININ GELİŞİMİ

Abonenin telefonunu OFF-HOOK yaptığı andan itibaren başlayıp ve aradığı abone ile bağlantının sağlandığı ana kadar devam eden süre içinde abone ile santral arasında gerçekleştirilen sinyalleşmelerin oluşum sürecine "call progress" adı verilir.

Abone telefonunu OFF-HOOK yaptığı anda santral, hat akımının yükselmesinden dolayı bunu anlar ve eğer boş bir register (Pulse veya DTMF sinyal alıcı) numara çevrilebileceğini işaret eden bir sinyal gönderir. Tepeden tepeye 2 V genliğinde 450 Hz frekansında olan bu sinyale "dial tone (çevir sesi)" adı verilir. Dial tonu duyan abone aradığı abonenin numarasını en ağırlıklı sayı hanesinden başlayarak çevirmeye başlar. İlk numarayı çevirmeye başladığı anda dial tonu kesilir ve böylece abone çevirdiği numaraların santral tarafından alındığını anlar. Girilen ilk numara eğer 0 ise santral aboneyi şehirlerarası hatta çıkartır ve farklı tonda ikinci bir çevir sesi ile bunu bildirir. Eğer ikinci bir 0 çevrilirse bu kez aboneyi milletlerarası hatta çıkartır ve üçüncü bir çevir

sesiyle de bunu bildirir. Numara çevirme bittikten sonra santral aranan abone ile bağlantı kurmaya çalışır.

Eğer aranan abone santralin kendi abonelerinden ise, santral hemen HOOK-SWITCH kontrolü yapar. OFF-HOOK konumunda ise arayan aboneye meşgul sinyali (busy tone) gönderir. Bu sinyal çevir sesi sinyalinin 0.5 saniye ON, 0.5 saniye OFF olacak şekilde bölünmesiyle elde edilir. Şayet aranan abone ON-HOOK konumunda ise hattına ring sinyali verilir. 70 Vefektif ve 25 Hz'lik ring sinyali aranan aboneye gönderilirken arayan aboneye de "ring back tone" olarak bildirilir. Ring back tone çevir sesinin gönderilen ring sinyali ile aynı sürelerde (2 saniye ON, 4 saniye OFF) arayan abone hattına verilmesi suretiyle gerçekleştirilir.

Eğer numara santralin kendi abonelerinden biri değilse (trank hattı bağlantısı) numaranın bağlı olduğu santral ile bağlantı kurmaya çalışır. Eğer boş hat bulamazsa arayan aboneye hat meşgul sinyali gönderir. Bu sinyal çevir sesi sinyalinin (3 kısa 1 uzun) olacak şekilde bölünmesiyle elde edilir. Boş kanal bulunduğu santraller arası bağlantı sağlanır ve bir önceki paragrafta anlatılmış olan işlemler aranan abonenin bağlı olduğu santral tarafından gerçekleştirilir.

5.3 TELEFON HATLARINDA KORUMA

Telefon hatlarında oluşabilecek şebeke gerilimi ve yıldırımlardan telefon, faks, modem gibi elektronik cihazların nasıl ve hangi yöntemlerle korunabileceğini inceleyeceğiz.

Telefon hatlarında 45-48 V DC ON-HOOK gerilimi, 60-80 V AC RING gerilimi bulunabilmektedir. Bu gerilim değerlerinin üzerinde oluşabilecek AC & DC gerilimlerin hatta bağlı elektronik cihaza geçmemesi için hat ile cihaz arasında bir koruyucu devre konulur. Bu devre maliyetine ve amacına göre 5 değişik şekilde oluşturulabilir.

5.3.1 Varistör

Varistör bir çeşit dirençtir. Uçlarına uygulanan voltaja bağlı olarak direnci değişmektedir. Herhangi bir sebeple hat voltajı yükselirse varistör 0.01 ms mertebesinde bir sürede tepki gösterir (reaction time), yavaş hızdadır. Dayanma süresi gücüne bağlı olarak 1 saniyeye kadar çıkabilir ve sürekli şebeke gerilimi altında bozulur. Avantajlı tek yanı ise en ucuz koruma şekli oluşudur.

5.3.2 Zener Diyod

İki adet 180 V'luk zener diyod ve bunlara seri bağlı küçük değerlikli bir direnç (10 Ω) ile oluşturulmuş koruma devresidir. Eğer hat voltajı, herhangi bir sebeple yükselir ve zener gerilimine ulaşırsa 100 ns ile 1 μ s arasında iletme geçerek aşırı voltajı 180 V'da sabit tutup kendi üzerinden akıtır. Bu esnada geçen aşırı akım, seri direncin belli bir süre sonra yanarak bozulmasına sebep olur. Direnç yanınca elektronik devre hattın tamamen yalıtılmış olur.

Seri direnç için 5-10 Ω arasında, 1/8 veya 1/4 W gücünde ve metal film direnç kullanılmalıdır. Eğer karbon türü bir direnç kullanılırsa, direnç yanarken alev alabilir, bu da baskı devreyi yakabilir.

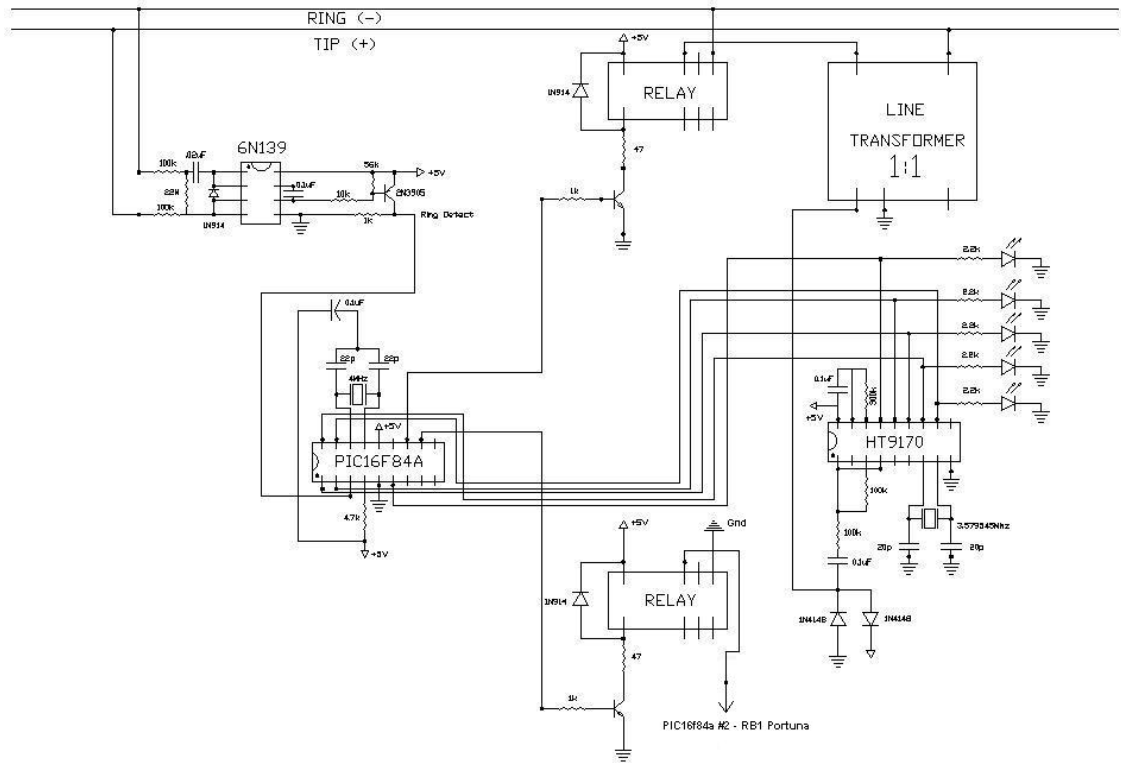
Zenerler, 1-2 W gücünde olmalı ve arka arkaya (back to back) bağlanmalıdır.

5.3.3 Gas Arrester

Telefon devresinde kullanılan gas arresterlar 200 V'luk seçilir. Eğer hat voltajı bu belirlenen voltajın üzerine çıkarsa gazın özelliğinden ötürü gas arrester deşarj olur ve üzerinden ters yönde bir desarj akımı akar. Bu akım hat voltajını düşürür. Tepki zamanı 1 μ s'dir. Gas arresterın uzun süreli yüksek gerilimlerde bozulmaması için, devreye küçük değerlikli, akım sınırlayıcı bir metal-film direnç, seri olarak ilave edilir. Gas arresterlar,

uçlarındaki gerilim V_{Br} gerilimini aşmadığı sürece açık devre gibi davranır ve zenerlerden daha dayanıklı olup daha büyük güçleri soğuracaktır.

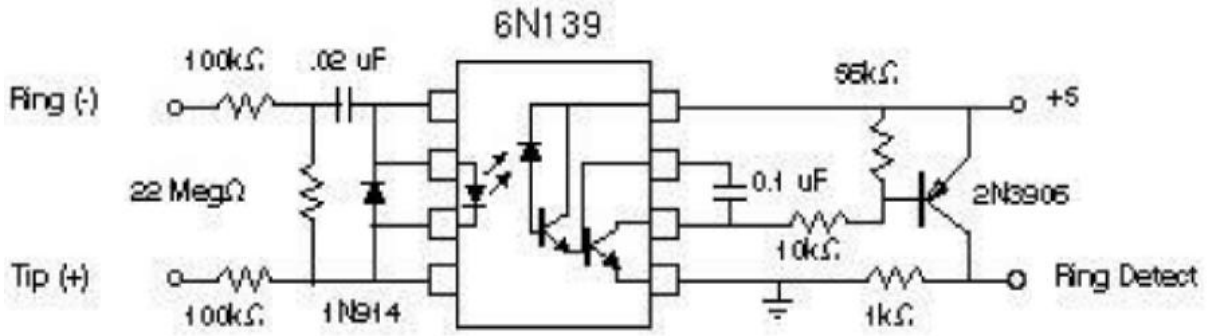
Gas arresterlar yıldırım korumasının önemli olduğu yerlerde (telefon santralleri gibi) pahalı olmalarına rağmen kullanılırlar.



Şekil 5.1: Telefon Hattından Kontrol Devresi

5.4 TELEFON RING DEVRESİ

Sayısal girişleri olan bir ev otomasyonu sisteminde kullanıldığında bu devre, telefon hattına çağrı geldiğinde sinyal üretir. Çoklu telefon hatlarında, her bir hat için bu devreden bir adet kullanılarak hangi hatta çağrı geldiği belirlenebilir.



Şekil 5.2: Telefon Ring Devresi

Telefon hattına çağrı geldiğinde, gerilim, 90 Volt ve 20 Hz'lik ve bazı zamanlar "Jingle Juice" olarak adlandırılan sinyal, hatta tespit edilir ve 100 k'lık dirençlerle sınırlandırılan çok küçük bir akım akıtır. .02 μF 'lık kapasite üzerinden akım bu küçük akım, 6N139 entegresi içindeki LED'i çalıştırır. 1N914 diyodu IR LED'i AC sinyalin negatif alternansından dolayı oluşacak olan akımdan korur. LED infrared ışık yaydığına, bu ışınlar fotodiyotu iletime geçirir ve çok küçük bir akım akıtır. Entegre devrenin içindeki amplifikatör devresinin ilk transistörü bu küçük akımı artırır ve ikinci transistörü doyuma ulaştırarak toprağa çeker. Sonunda, harici olarak bağlanan 2N3906 transistörü yükseltme işleminin son aşamasını gerçekleştirerek "Ring Detect" çıkışını lojik1 konumuna çeker. 0.1 μF 'lık kondansatör AC darbeleri filtre ederek çıkışa temiz bir lojik1 sinyalinin verilmesini sağlar.

Bu devre telefon hattına çağrı geldiğinde çıkışa lojik1 verir, fakat telefon hattını açabilecek kadar yeterli akımı çekemez.

5.5 DTMF

DTMF, çift tonlu çoklu frekans (Dual-Tone Multi-Frequency) demektir. DTMF işareti, değişik frekanslı aynı genlikli iki adet saf sinüs dalgasının toplamından oluşur. Örneğin telefonda “1” tuşuna basıldığında, 1209 Hz frekanslı ve 697 Hz frekanslı aynı genlikli iki adet sinüs dalgasının toplamından oluşan sinyal hattın diğer ucuna gönderilir.

5.5.1 DTMF Frekansları

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	<i>1</i>	<i>2</i>	<i>3</i>	<i>A</i>
770 Hz	<i>4</i>	<i>5</i>	<i>6</i>	<i>B</i>
852 Hz	<i>7</i>	<i>8</i>	<i>9</i>	<i>C</i>
941 Hz	<i>*</i>	<i>0</i>	<i>#</i>	<i>D</i>

Tablo 5.1: DTMF Frekansları

“4” tuşuna basıldığında, 770 Hz frekanslı ve 1209 Hz frekanslı işaretler toplanıp, birlikte gönderilir. Ardından telefon santrali bu ton çiftlerinden numarayı çözer.

Ton frekansları, gönderme ve alma işlemleri sırasında harmoniklerin ve diğer problemlerin oluşmasını engelleyecek şekilde tasarlanmıştır. İşaretin iletim doğruluğu çok önemlidir. Ton, nominalden +/- %1.5 kadar farklı olabilir. Yüksek frekanslı ton, en azından ve tavsiye edildiği için, düşük frekanslı tondan daha kuvvetli olmalıdır. 4 dB fazla olması yeterlidir.

Tuş takımındaki A, B, C ve D sinyalleri ilave işaretlerdir ve kökeni A.B.D.’nin askeri “Autovon Phone Network” isimli ağına dayanır. Bu tuşların orijinal isimleri ise; FO – Flash Override (Çok Yüksek Öncelikli), F – Flash (Yüksek Öncelikli), I – Immediate

(Acil) ve P – Priority (Öncelikli) şeklindedir. Bunlar telefon ağında gerçekleşen bağlantıların öncelik seviyelerini belirlemek için kullanılır.

Günümüzde A, B, C ve D sinyalleri genel olarak amatör radyo uygulamaları gibi özel uygulamalarda işaret kontrolünde kullanılır. Ayrıca modemlerde bu sinyalleri kullanırlar. Bu tonlar uzun bir süredir genel amaçlı olarak kullanılmıyor.

5.5.2 DTMF Kodlama

DTMF tonlarını kodlamak ve çözmek için çok çeşitli yollar mevcuttur. Temel prensibi 8 sinyal filtresinin değişik kombinasyonlarını kullanarak tasarlanan tarama-tespit devrelerdir. Bu hiç pratik bir yöntem değildir. Bunun yerine tüm dünyada değişik üreticiler tarafından üretilmiş olan entegre devreler kullanmak daha uygundur.

Bir çok DTMF entegre devresi, son derece ucuz olan, bir adet 3.58 MHz'lik kristal veya seramik rezonatör ile çalışırlar ve genellikle çıkışı 4-bit binary + 1 strobe olarak kodlarlar.

```
-----  
1          1- d3  
1      DTMF  1- d2  
signal in -1  Decoder  1- d1      4-bit binary out  
1      chip  1- d0  
1          1- strobe  
-----
```

Ayrıca DTMF sinyalleri için önerilen toleranslar şu şekildedir:

Frequency Tolerance:	Operation:	<= 1.5%
	Non-Operation:	>= 3.5%
Signal Duration:	Operation:	40ms max
	Non-Operation:	23ms min
Signal Interruption:		10ms max
Twist:	Normal:	8db max
	Reverse:	4db max

5.6 ÖLÇÜM SONUÇLARI

ON-HOOK Durumunda: -16,3 V (DC)

OFF-HOOK Durumunda: -7,4 V (DC)

Ring Sinyali: $V_{\text{eff}} = 58$ V (Dijital Multimetre ile ölçülmüştür)

Bu ölçümler Yıldız Teknik Üniversitesi'nin dahili telefon santralinden elde edilmiştir. Bu yüzden standart değerlerden farklı olabilir.

Ayrıca telefon hattında çok fazla gürültü olabileceği için toleranslar büyüktür. Bu, santralin analog veya dijital olmasına da bağlıdır.

5.7 MODÜLÜN ÇALIŞMA PRENSİBİ

Ring devresi, telefon çalarken $V_{\text{pick}}=162$ V ve $f=20-25$ Hz'lik AC sinyali dijital sinyale çevirir. Bu devreden alınan dijital çıkış mikrokontrolöre girilir ve telefon belli bir sayıda çaldıktan sonra yine mikrokontrolör vasıtasıyla telefon açma-kapama devresi kontrol edilir.

Telefon açma-kapama devresinin kontrol ucuna lojik1 geldiğinde hat açılır ve telefon hattından 20-80 mA'lık bir akım akması sağlanır. Mikrokontrolörün çıkışından 20 mA civarında bir akım akabilir. Bu 20 mA'lık akım rölenin açma-kapama yapabilmesi için yeterli değildir. Bu yüzden mikrokontrolörün lojik çıkışı, β_{dc} kazancı yaklaşık 100 olan npn tipi bir transistörün “baz” girişine verilerek, röleden 40 mA civarında bir akım akması sağlanmıştır. Röleden 20-80 mA'lık bir akım aktığında, telefon hattı açılmış olur.

Yalıtım trafosu devreyi manyetik olarak telefon hattından izole eder. Böylelikle hatta bulunan DC gerilimden devre korunmuş olur. Telefon çalarken gelen AC sinyali de izole etmek için yalıtım trafosu röle çıkışına bağlandı. Bu şekilde HT9170 DTMF dekoder entegre devresine yalnızca DTMF sinyallerinin iletilmesi sağlandı.

Hat açıldıktan sonra telefon tuş takımı ile gönderilen DTMF sinyalleri, HT9170 entegre devresi vasıtasıyla 4+1 bit şeklinde dijital olarak kodlandı. İlk dört bit tuş takımından basılan tuşun dijital olarak kodlanmasıdır. Bu dört bit tutma (latch) konumundadır. Bu yüzden 5. bir daha kullanılmıştır. Bu bit “strobe” çıkışı olarak adlandırılır. Ve yalnızca bir tuşa basılırken lojik1 komunda kalır. Böylelikle aynı tuşa kaç defa basıldığı veya başka bir tuşa basıldığı kontrol edilebilir.

Daha sonra HT9170 entegresinden alınan 4+1 bitlik lojik çıkışlar mikrokontrolöre girilir ve gelen dijital bilgiye göre istenilen çıkışlar değiştirilebilir.

Devrede mikrokontrolör olarak PIC16F84A kullanılmıştır. Bunun nedeni bu mikrokontrolörün çok yaygın olarak kullanılması ve bu sebepten dolayı da ucuz olmasıdır. Böylelikle proje maliyeti minimize edilmeye çalışılmıştır.

5.8 KAYNAK KODLAR

```
DEFINE OSC 4          ' Osilatör frekansını 4MHz olarak ayarla
role_say var byte     '
TRISA=%11111111      ' A portu çıkış
TRISB=%00000001     ' B portu çıkış (1. biti giriş)
PORTA=0              ' A portu sıfırla
PORTB=0
role_say=0
Main:                 ' Main etiketi
  if PORTA.4=1 then
  PAUSE 20
  IF PORTA.4=0 and role_say<5 THEN
  role_say=role_say+1
  ENDIF
  ENDIF
```

```
IF role_say=5 then
PORTB.6=1
PAUSE 100
IF PORTA.0=0 and PORTA.1=0 AND PORTA.2=0 and PORTA.3=1 THEN
PORTB.5=1
ENDIF

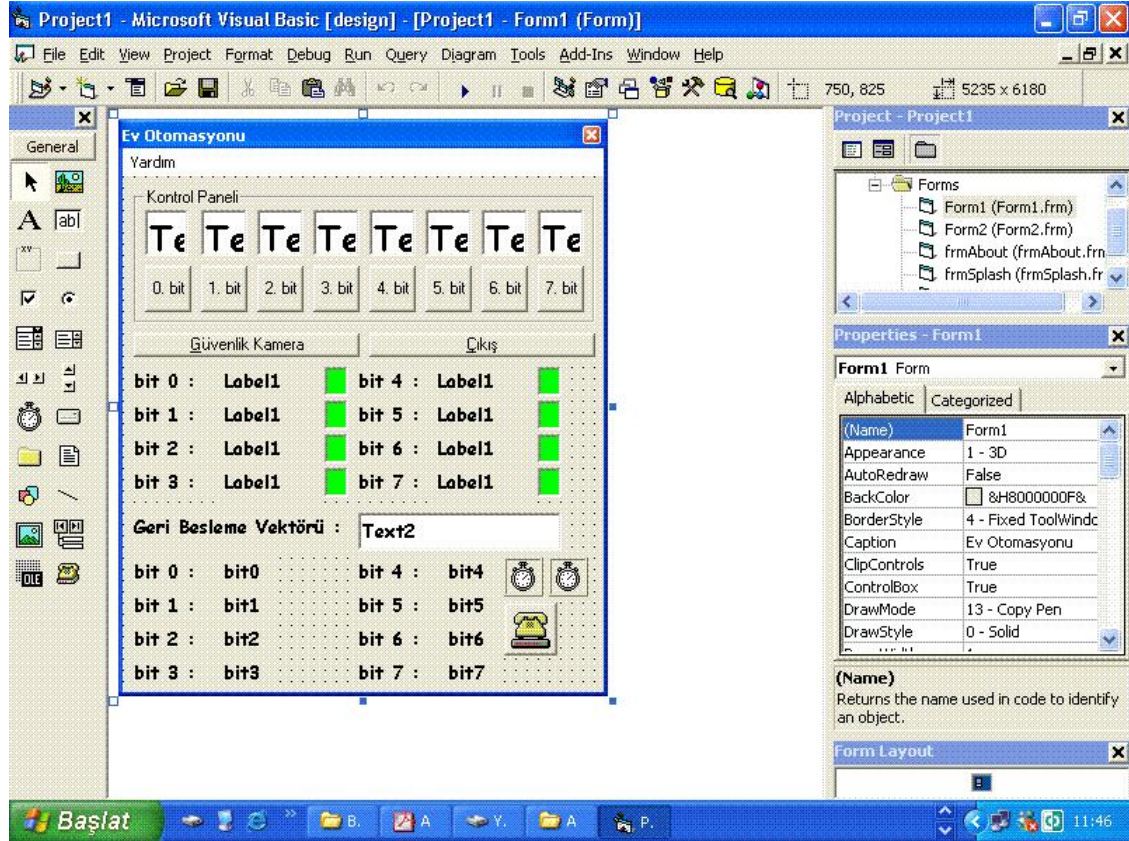
IF PORTA.0=0 and PORTA.1=0 AND PORTA.2=1 and PORTA.3=0 THEN
PORTB.5=0
ENDIF

IF PORTA.0=1 and PORTA.1=0 AND PORTA.2=0 and PORTA.3=0 and
PORTB.0=1 THEN
PORTB.6=0
role_say=0
ENDIF

ENDIF
GOTO MAIN          ' Main etiketine geri dön

END                ' Programı bitir
```

BÖLÜM 6: GRAFİK KULLANICI ARAYÜZÜ



Şekil 6.1: Form1.frm

Form1.frm

Public b7 As String

Public b6 As String

Public b5 As String

Public b4 As String

Public b3 As String

Public b2 As String

Public b1 As String

Public b0 As String

Private Sub Command1_Click(Index As Integer)

```
If txtport(Index) = 1 Then  
txtport(Index) = 0  
Else  
txtport(Index) = 1  
End If
```

```
'-----  
Kontrol = ""  
For i = 7 To 0 Step -1  
Kontrol = Kontrol & txtport(i)  
Next  
Kontrol = "P" & Kontrol  
myCom.Output = Kontrol  
'-----
```

```
End Sub
```

```
Private Sub Command2_Click()  
Kontrol = ""  
For i = 7 To 0 Step -1  
Kontrol = Kontrol & txtport(i)  
Next  
Kontrol = "P" & Kontrol  
myCom.Output = Kontrol  
End Sub
```

```
Private Sub Command4_Click()  
Form1.Hide  
Form2.Show  
End Sub
```

```
Private Sub Command6_Click()  
bit7.Caption = b7  
bit6.Caption = b6  
bit5.Caption = b5  
bit4.Caption = b4  
bit3.Caption = b3  
bit2.Caption = b2  
bit1.Caption = b1  
bit0.Caption = b0  
End Sub
```

```
Private Sub Form_Load()  
'-----  
For i = 0 To 7  
txtport(i) = 1  
Next  
myCom.CommPort = 1  
myCom.Settings = "9600,n,8,1"  
myCom.PortOpen = True  
'-----  
'com1'i ac, gerekli ayarlamalari yap  
'butun kontrol bitlerini logic1 yap  
'-----  
  
Form1.Hide  
frmSplash.Show  
  
'-----  
Kontrol = ""  
For i = 7 To 0 Step -1  
Kontrol = Kontrol & txtport(i)
```

```
Next
Kontrol = "P" & Kontrol
myCom.Output = Kontrol
'-----
'kontrol vektorunu mikrokontrole ata
'bu durumda butun kontrol bitleri logic1 olur
'-----
```

```
End Sub
```

```
Private Sub Command3_Click()
myCom.Output = "P00000000"
myCom.PortOpen = False
End
End Sub
```

```
Private Sub Hakkında_Click(Index As Integer)
frmAbout.Show
End Sub
```

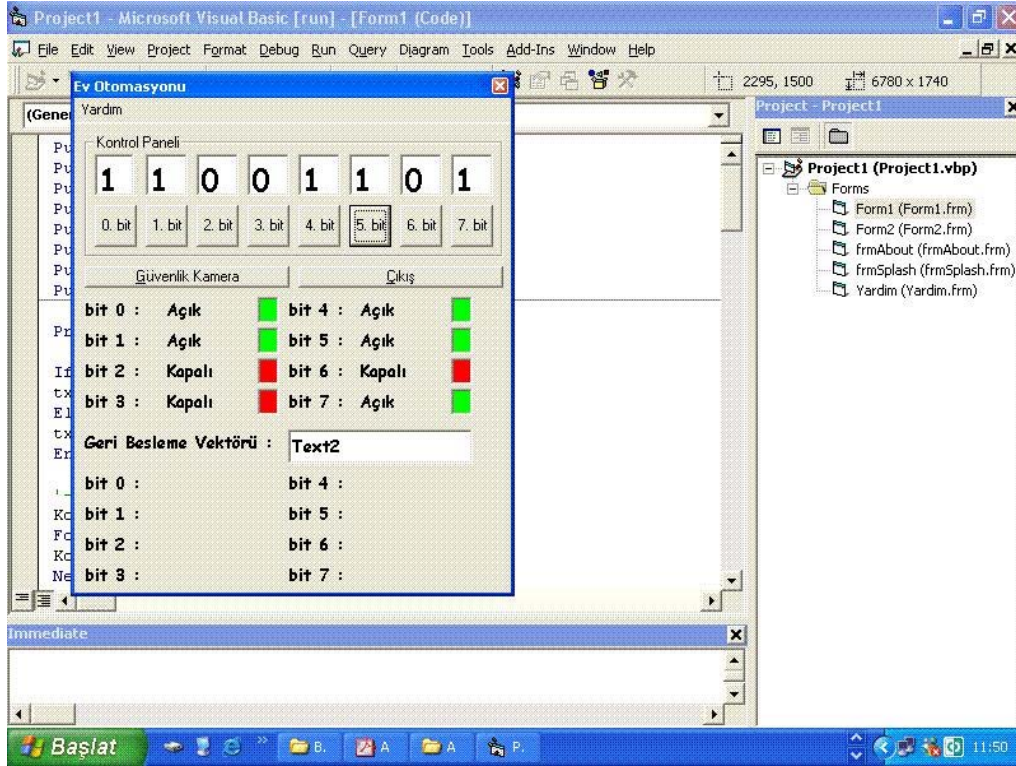
```
Private Sub myCom_OnComm()
If myCom.Input = "P" Then
b7 = myCom.Input
b6 = myCom.Input
b5 = myCom.Input
b4 = myCom.Input
b3 = myCom.Input
b2 = myCom.Input
b1 = myCom.Input
b0 = myCom.Input
```

```
getData.Text = b7 & b6 & b5 & b4 & b3 & b2 & b1 & b0  
End If  
End Sub
```

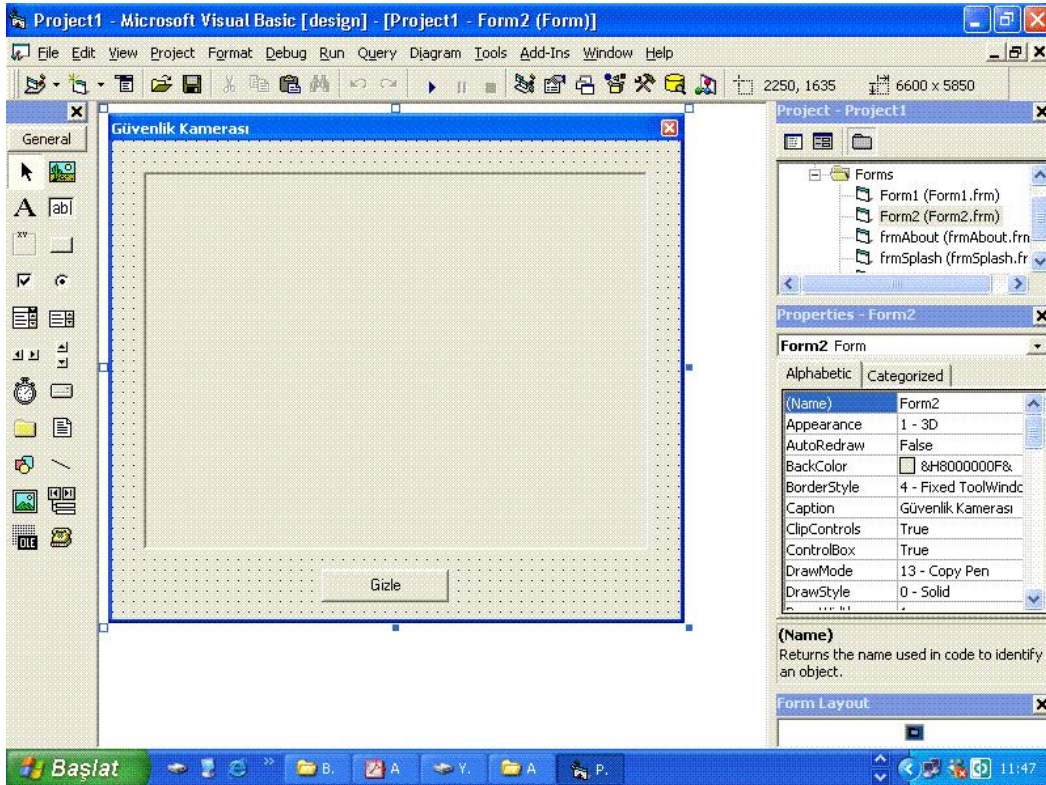
```
Private Sub Timer1_Timer()  
For i = 0 To 7 Step 1  
If txtport(i).Text = "0" Then  
Label1(i).Caption = "Kapalı"  
Text1(i).BackColor = &HFF&  
Else  
Label1(i).Caption = "Açık"  
Text1(i).BackColor = &HFF00&  
End If  
Next i  
End Sub
```

```
Private Sub Timer2_Timer()  
bit7.Caption = b7  
bit6.Caption = b6  
bit5.Caption = b5  
bit4.Caption = b4  
bit3.Caption = b3  
bit2.Caption = b2  
bit1.Caption = b1  
bit0.Caption = b0  
End Sub
```

```
Private Sub Yardim_Iste_Click(Index As Integer)  
Yardim.Show  
End Sub
```

Şekil 6.2: Form1



Şekil 6.3: Form2.frm

Form2.frm

```
Private Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal  
lpClassName As Long, ByVal lpWindowName As Long) As Long  
Private Declare Function GetParent Lib "user32" (ByVal hwnd As Long) As Long  
Private Declare Function SetParent Lib "user32" (ByVal hWndChild As Long, ByVal  
hWndNewParent As Long) As Long  
Private Declare Function GetWindowThreadProcessId Lib "user32" (ByVal hwnd As  
Long, lpdwProcessId As Long) As Long  
Private Declare Function GetWindow Lib "user32" (ByVal hwnd As Long, ByVal wCmd  
As Long) As Long  
Private Declare Function LockWindowUpdate Lib "user32" (ByVal hwndLock As Long)  
As Long  
Private Declare Function GetDesktopWindow Lib "user32" () As Long  
Private Declare Function DestroyWindow Lib "user32" (ByVal hwnd As Long) As Long  
Private Declare Function TerminateProcess Lib "kernel32" (ByVal hProcess As Long,  
ByVal uExitCode As Long) As Long  
Private Declare Function GetCurrentProcess Lib "kernel32" () As Long  
Private Declare Function Putfocus Lib "user32" Alias "SetFocus" (ByVal hwnd As Long)  
As Long  
Const GW_HWNDNEXT = 2  
Dim mWnd As Long
```

```
Function InstanceToWnd(ByVal target_pid As Long) As Long
```

```
    Dim test_hwnd As Long, test_pid As Long, test_thread_id As Long  
    test_hwnd = FindWindow(ByVal 0&, ByVal 0&)
```

```
    Do While test_hwnd <> 0
```

```
        If GetParent(test_hwnd) = 0 Then
```

```
test_thread_id = GetWindowThreadProcessId(test_hwnd, test_pid)
```

```
    If test_pid = target_pid Then  
        InstanceToWnd = test_hwnd  
        Exit Do  
    End If  
End If
```

```
test_hwnd = GetWindow(test_hwnd, GW_HWNDNEXT)
```

```
Loop
```

```
End Function
```

```
Private Sub Command1_Click()
```

```
Form2.Hide
```

```
Form1.Show
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Dim Pid As Long
```

```
    LockWindowUpdate GetDesktopWindow
```

```
    'Buraya çalışmasını istediğiniz programı yazın
```

```
    Pid = Shell("kamera.exe", vbNormalFocus)
```

```
    If Pid = 0 Then MsgBox "Error starting the app"
```

```
    mWnd = InstanceToWnd(Pid)
```

SetParent mWnd, Picture1.hwnd 'PİCTURE1 İÇİNDE ÇALIŞTIRIYO

'SetParent mWnd, Me.hwnd 'FORM İÇİNDE ÇALIŞTIRIYO

Putfocus mWnd

LockWindowUpdate False

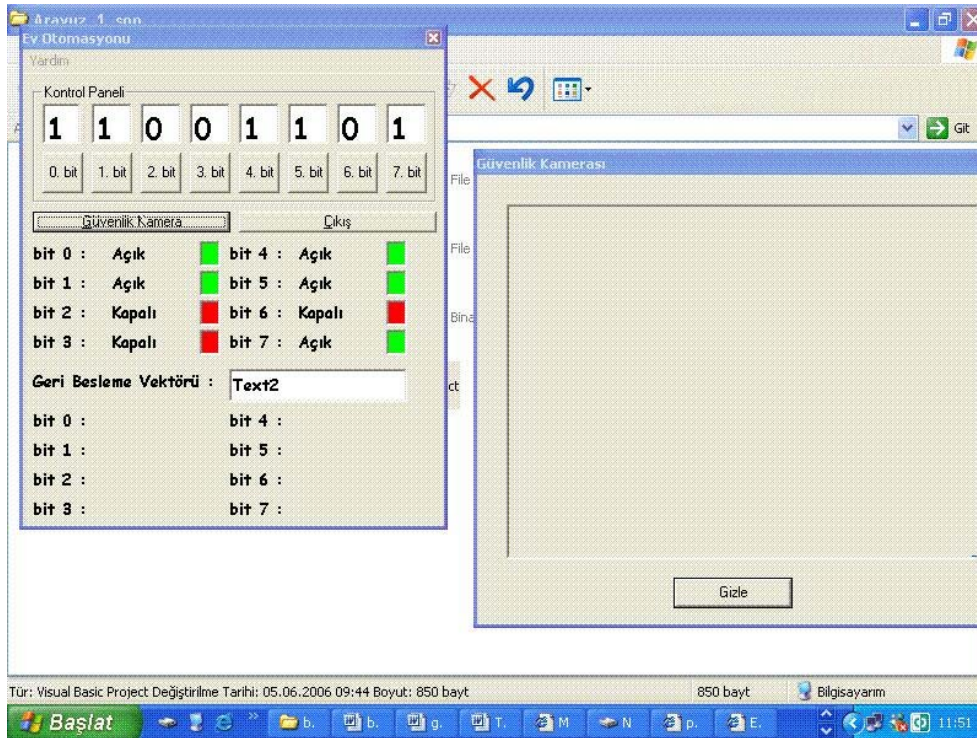
End Sub

Private Sub Form_Unload(Cancel As Integer)

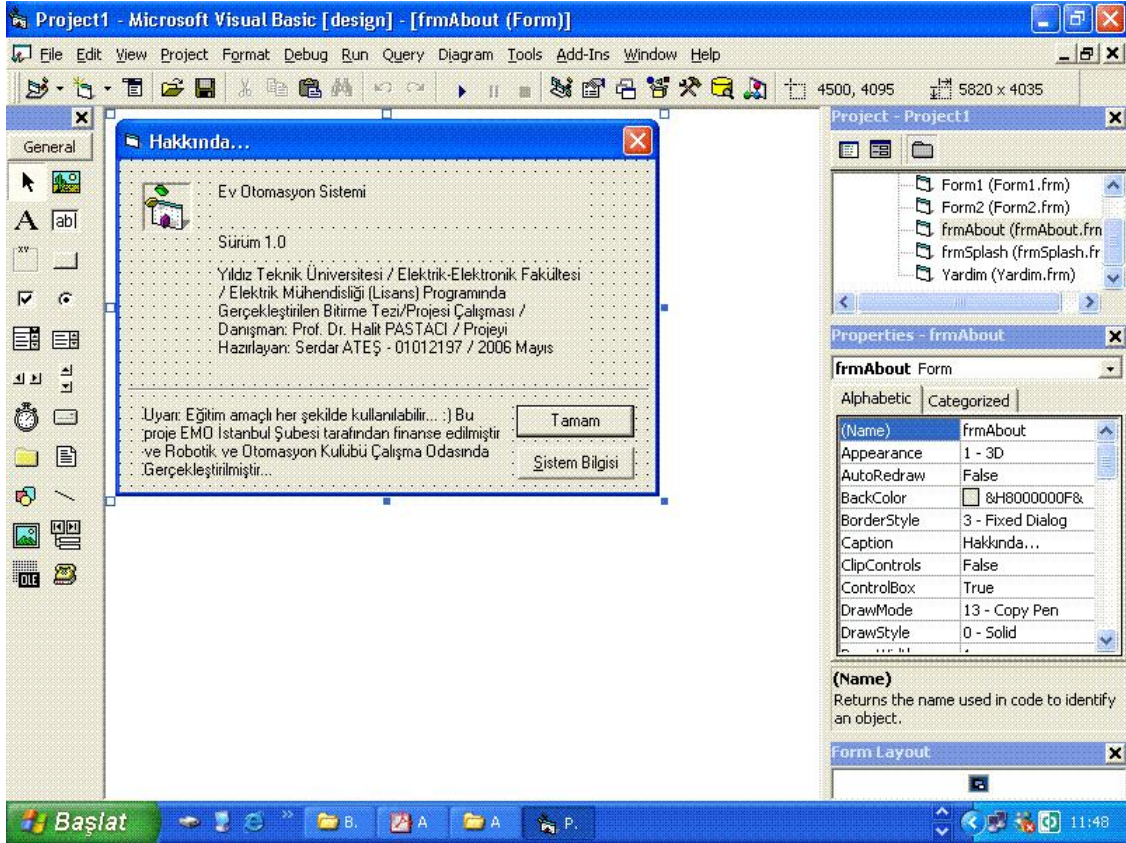
'Çalışan program kapatılıyor

DestroyWindow mWnd

End Sub



Şekil 6.4: Form2



Şekil 6.5: frmAbout.frm

frmAbout.frm

Option Explicit

' Reg Key Security Options...

Const READ_CONTROL = &H20000

Const KEY_QUERY_VALUE = &H1

Const KEY_SET_VALUE = &H2

Const KEY_CREATE_SUB_KEY = &H4

Const KEY_ENUMERATE_SUB_KEYS = &H8

Const KEY_NOTIFY = &H10

Const KEY_CREATE_LINK = &H20

Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE + _
KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS + _
KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL

```
' Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1           ' Unicode nul terminated string
Const REG_DWORD = 4       ' 32-bit number

Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA"
(ByVal hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal
samDesired As Long, ByRef phkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias "RegQueryValueExA"
(ByVal hKey As Long, ByVal lpValueName As String, ByVal lpReserved As Long,
ByRef lpType As Long, ByVal lpData As String, ByRef lpcbData As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As Long

Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

Private Sub Form_Load()
    Me.Caption = "About " & App.Title
```

```
'lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." &  
App.Revision
```

```
'lblTitle.Caption = App.Title
```

```
End Sub
```

```
Public Sub StartSysInfo()
```

```
On Error GoTo SysInfoErr
```

```
Dim rc As Long
```

```
Dim SysInfoPath As String
```

```
' Try To Get System Info Program Path\Name From Registry...
```

```
If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO,  
gREGVALSYSINFO, SysInfoPath) Then
```

```
' Try To Get System Info Program Path Only From Registry...
```

```
ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC,  
gREGVALSYSINFOLOC, SysInfoPath) Then
```

```
' Validate Existence Of Known 32 Bit File Version
```

```
If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
```

```
    SysInfoPath = SysInfoPath & "\MSINFO32.EXE"
```

```
' Error - File Can Not Be Found...
```

```
Else
```

```
    GoTo SysInfoErr
```

```
End If
```

```
' Error - Registry Entry Can Not Be Found...
```

```
Else
```

```
    GoTo SysInfoErr
```

```
End If
```

```
Call Shell(SysInfoPath, vbNormalFocus)
```

```
Exit Sub
SysInfoErr:
    MsgBox "System Information Is Unavailable At This Time", vbOKOnly
End Sub

Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef As
String, ByRef KeyVal As String) As Boolean
    Dim i As Long                ' Loop Counter
    Dim rc As Long              ' Return Code
    Dim hKey As Long           ' Handle To An Open Registry Key
    Dim hDepth As Long        '
    Dim KeyValType As Long     ' Data Type Of A Registry Key
    Dim tmpVal As String       ' Temporary Storage For A Registry Key
Value
    Dim KeyValSize As Long     ' Size Of Registry Key Variable
    '-----
    ' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}
    '-----
    rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' Open
Registry Key

    If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError    ' Handle Error...

    tmpVal = String$(1024, 0)                ' Allocate Variable Space
    KeyValSize = 1024                        ' Mark Variable Size

    '-----
    ' Retrieve Registry Key Value...
    '-----
    rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
```



```
KeyValType, tmpVal, KeyValSize) ' Get/Create Key Value

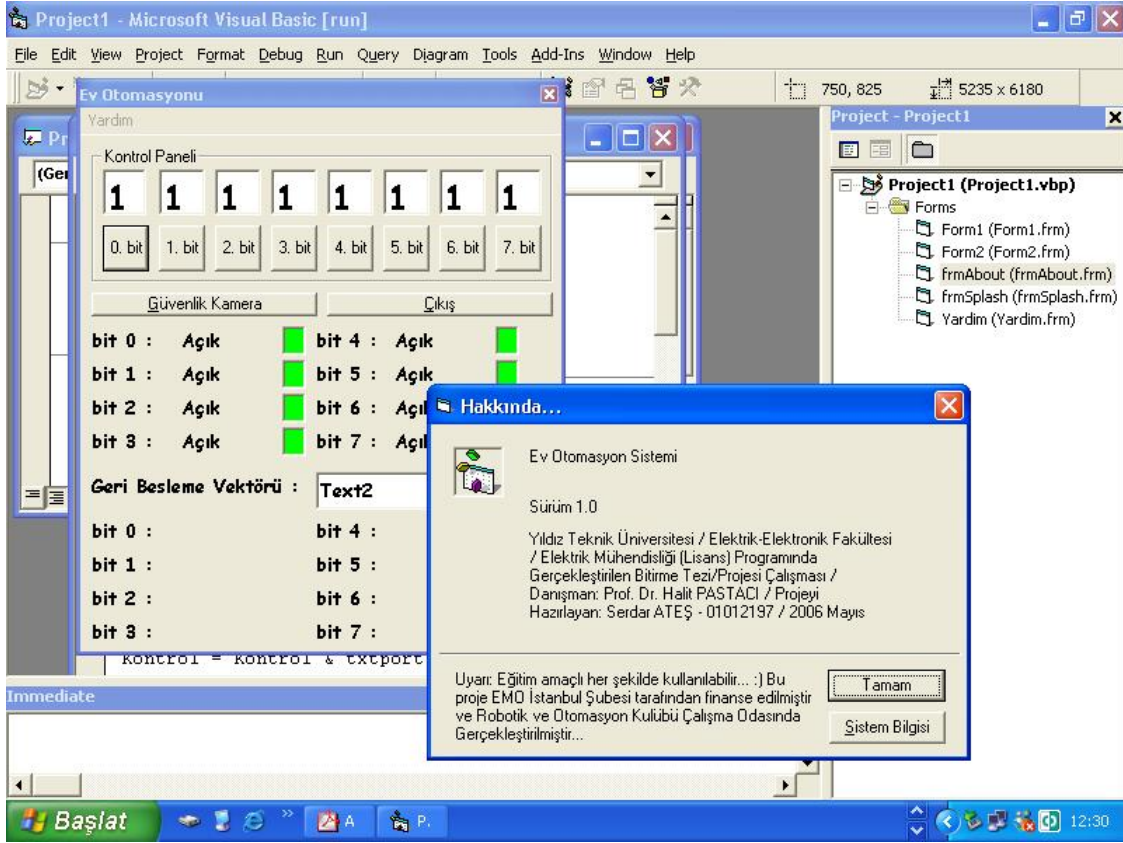
If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError      ' Handle Errors

If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then      ' Win95 Adds Null Terminated
String...
    tmpVal = Left(tmpVal, KeyValSize - 1)         ' Null Found, Extract From String
Else                                              ' WinNT Does NOT Null Terminate String...
    tmpVal = Left(tmpVal, KeyValSize)             ' Null Not Found, Extract String
Only
End If
'-----
' Determine Key Value Type For Conversion...
'-----
Select Case KeyValType                          ' Search Data Types...
Case REG_SZ                                     ' String Registry Key Data Type
    KeyVal = tmpVal                             ' Copy String Value
Case REG_DWORD                                 ' Double Word Registry Key Data Type
    For i = Len(tmpVal) To 1 Step -1            ' Convert Each Bit
        KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1))) ' Build Value Char. By Char.
    Next
    KeyVal = Format("&h" + KeyVal)                ' Convert Double Word To String
End Select

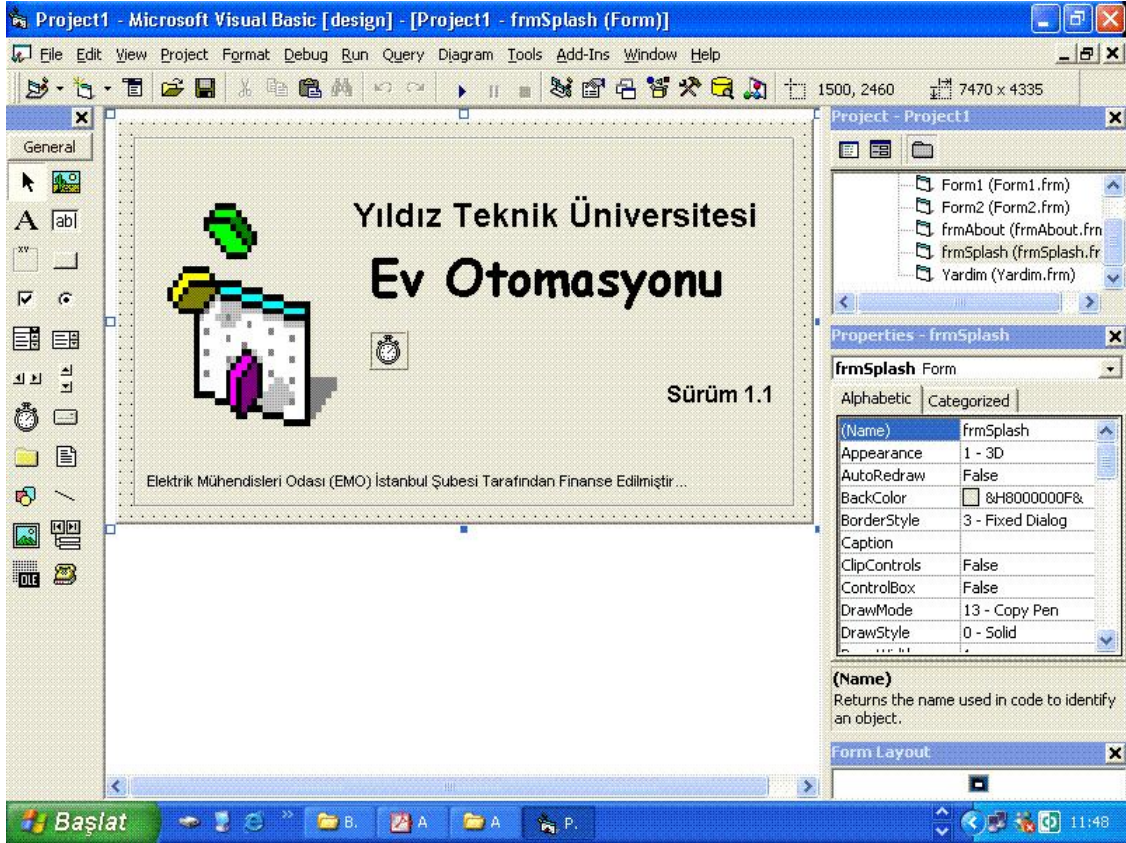
GetKeyValue = True                             ' Return Success
rc = RegCloseKey(hKey)                          ' Close Registry Key
Exit Function                                   ' Exit

GetKeyError:  ' Cleanup After An Error Has Occured...
    KeyVal = ""                                  ' Set Return Val To Empty String
    GetKeyValue = False                          ' Return Failure
```

```
rc = RegCloseKey(hKey)           ' Close Registry Key  
End Function
```



Şekil 6.6: frmAbout



Şekil 6.7: frmSplash.frm

frmSplash.frm

Option Explicit

```
Private Sub Form_KeyPress(KeyAscii As Integer)
```

```
    Unload Me
```

```
    Form1.Show
```

```
End Sub
```

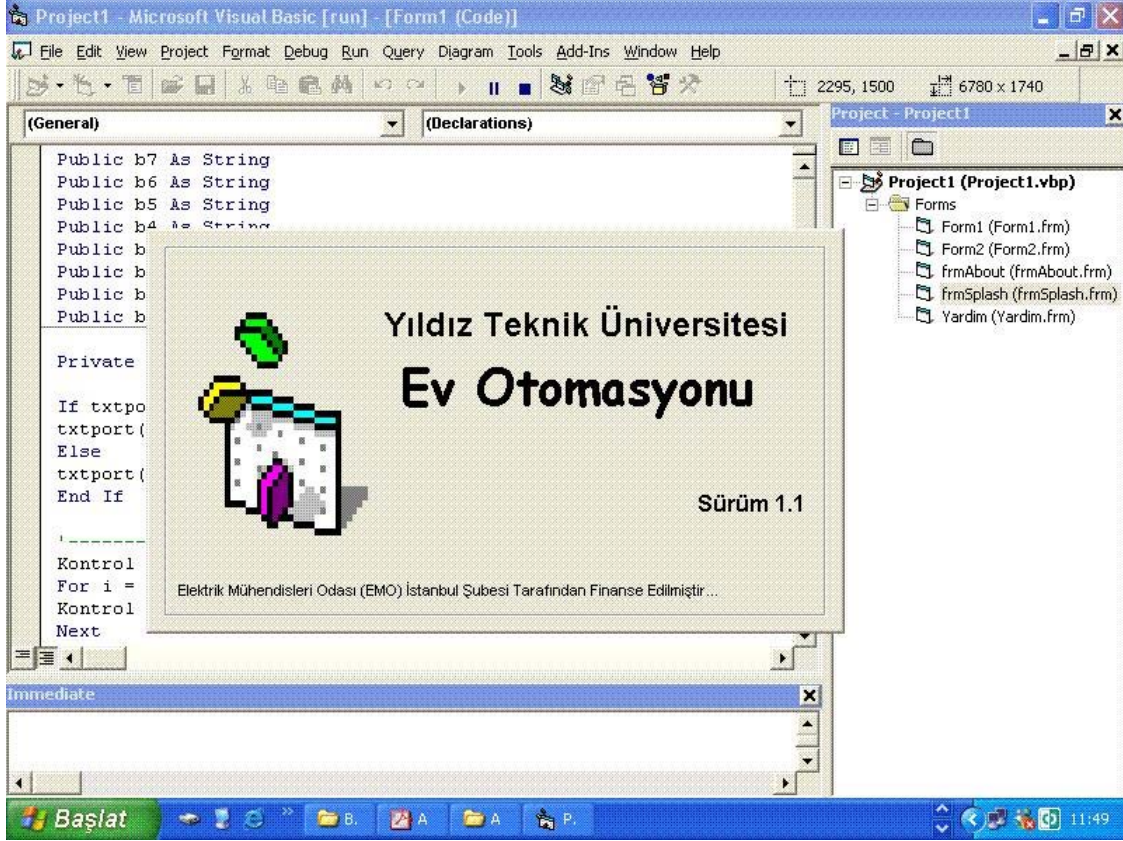
```
Private Sub Frame1_Click()
```

```
    Unload Me
```

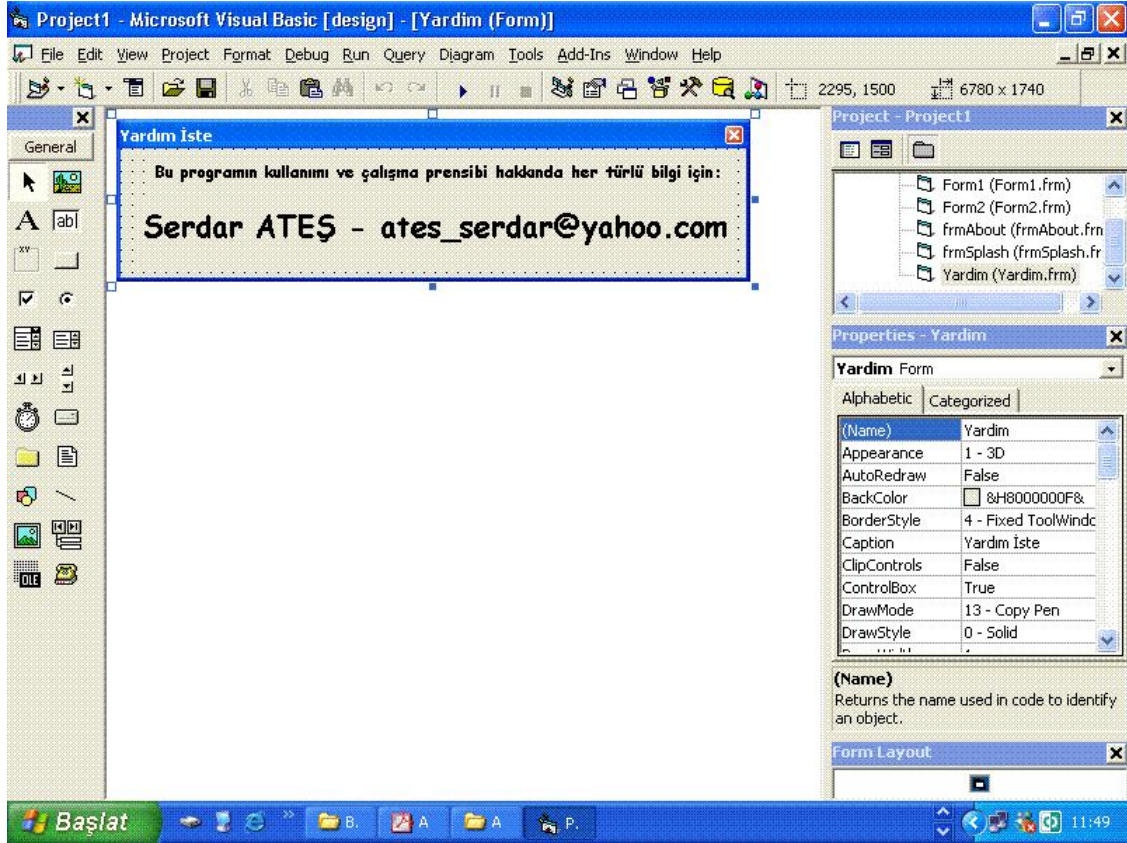
```
    Form1.Show
```

```
End Sub
```

```
Private Sub Timer1_Timer()  
frmSplash.Hide  
Form1.Show  
End Sub
```



Şekil 6.8: frmSplash



Şekil 6.9: Yardim.frm

Yardim.frm

```
Private Sub Form_Load()
```

```
End Sub
```

```
Private Sub Label1_Click()
```

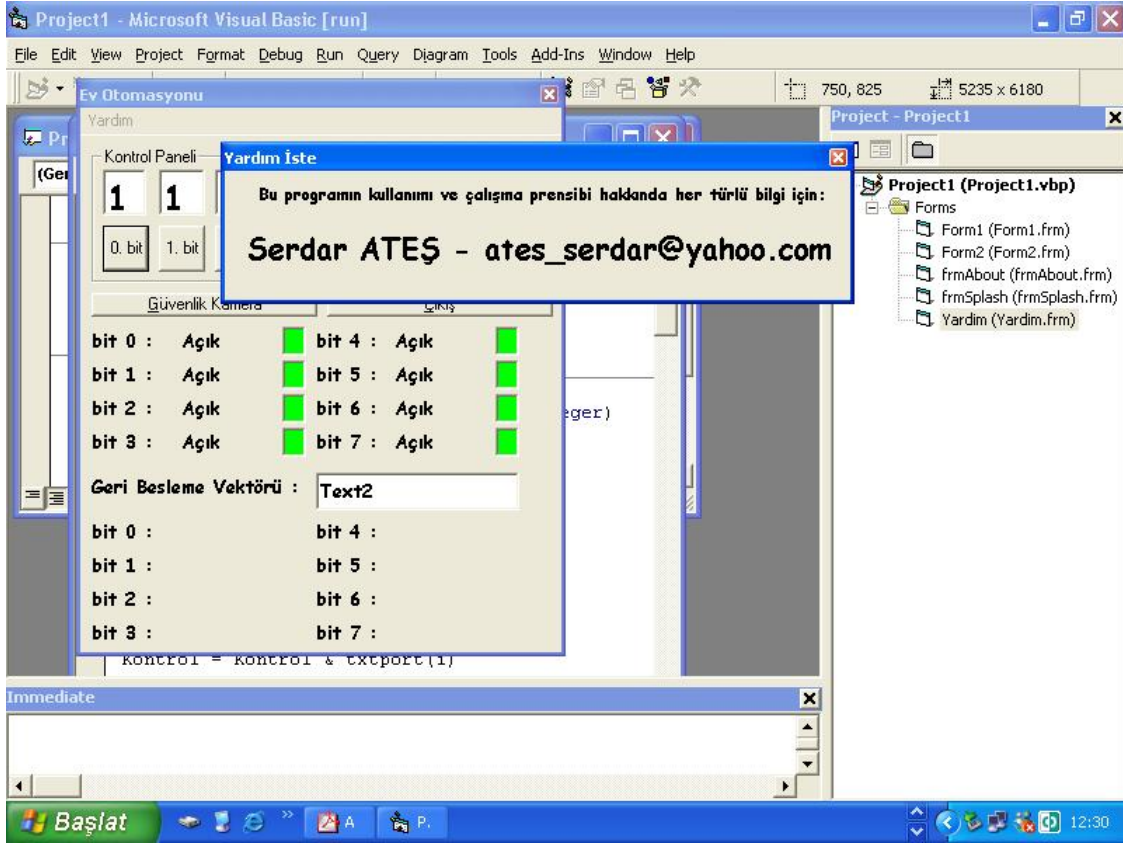
```
Yardim.Hide
```

```
End Sub
```

```
Private Sub Label2_Click()
```

```
Yardim.Hide
```

```
End Sub
```



Şekil 6.10: Yardım

BÖLÜM 7: YAZILIMLAR

Bu kısımda 8-bit aç/kapa kontrolünü gerçekleştiren mikrodenetleyici ile 8-bit geribesleme işlemini gerçekleştiren mikrodenetleyicinin yazılımı incelenmiştir. Bu yazılımın kolay ve kısa oluşu sebebiyle C programlama dili kullanılarak gerçekleştirilmesi uygun görülmüştür.

7.1 Kontrol Yazılımı

Bu mikrodenetleyici yazılımında PORTB'ye ait 8-bit çıkış, PORTA'nın 0.biti de giriş olarak tanımlanmıştır. PORTA'nın 0.biti MAX232 entegre devresi ile bilgisayarın seri portuna bağlanmıştır. Bilgisayarda yüklü olan grafik kullanıcı arayüzü tarafından

'P00001111' şeklinde bir patern gönderilmektedir. Buradaki 'P' karakteri parite karakteri olarak kullanılmaktadır. Mikrodenetleyicinin yazılım sürekli PORTA'nın 0.bitini okumaktadır ve 'P' karakterinin gelip gelmediğini kontrol etmektedir. Eğer 'P' karakteri gelmezse bu karakter gelene kadar beklemektedir. 'P' karakteri geldiği andan itibaren okuduğu ilk 8 karakteri (bunların tümü 0 ve 1'lerden meydana gelmektedir) PORTB'nin çıkışları olarak atar. Örneğin 'P00001111' paterni geldiğinde PORTB'nin ilk 4 biti lojik0, son 4 biti de lojik1 konumuna gelmektedir.

```
#include <16F84A.h>
```

```
#FUSES NOWDT                //No Watch Dog Timer
#FUSES
#use delay(clock=4000000)
#use rs232(baud=9600,parity=N,xmit=PIN_A1,rcv=PIN_A0,bits=8)
void main()
{
  unsigned char c,i,s;
  setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
  set_tris_a ( 0x01 );
  set_tris_b ( 0x00 );
  output_b ( 0x00 );
  while (1)
  {
    c = getc();
    if( c=='P' )
    {
      s = 0;
      for (i=0;i<8;i++)
      {
        c = getc();
        c = c - '0';
```

```
    s = (s<<1) + c;
  }
  output_b ( s );
}
}
```

7.2 Geri Besleme Yazılımı

Bu mikrodenetleyici yazılımında PORTB'ye ait 8-bit giriş, PORTA'nın 1.biti de giriş olarak tanımlanmıştır. PORTA'nın 1.biti MAX232 entegre devresi ile bilgisayarın seri portuna bağlanmıştır. Bilgisayarda yüklü olan grafik kullanıcı arayüzüne 'P00001111' şeklinde bir patern gönderilmektedir. Buradaki 'P' karakteri parite karakteri olarak kullanılmaktadır. Grafik kullanıcı arayüz yazılımı sürekli PORTA'nın 1.bitinden seri bilgiyi okumaktadır ve 'P' karakterinin gelip gelmediğini kontrol etmektedir. Eğer 'P' karakteri gelmezse bu karakter gelene kadar beklemektedir. 'P' karakteri geldiği andan itibaren okuduğu ilk 8 karakteri (bunların tümü 0 ve 1'lerden meydana gelmektedir) okuyup, kendi içinde çeşitli değişkenlere atamaktadır. Bu şekilde grafik kullanıcı arayüzündeki gerekli yazılım değişiklikleri yapılarak gelen geribesleme bilgisi istenildiği şekilde işlenebilmektedir.

```
#include <16F84A.h>
#FUSES NOWDT           //No Watch Dog Timer
#FUSES
#use delay(clock=4000000)
#use rs232(baud=9600,parity=N,xmit=PIN_A1,rcv=PIN_A0,bits=8)

void main()
{
  unsigned char c,i,s;
```



```
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
set_tris_a ( 0x01 );
set_tris_b ( 0xFF );

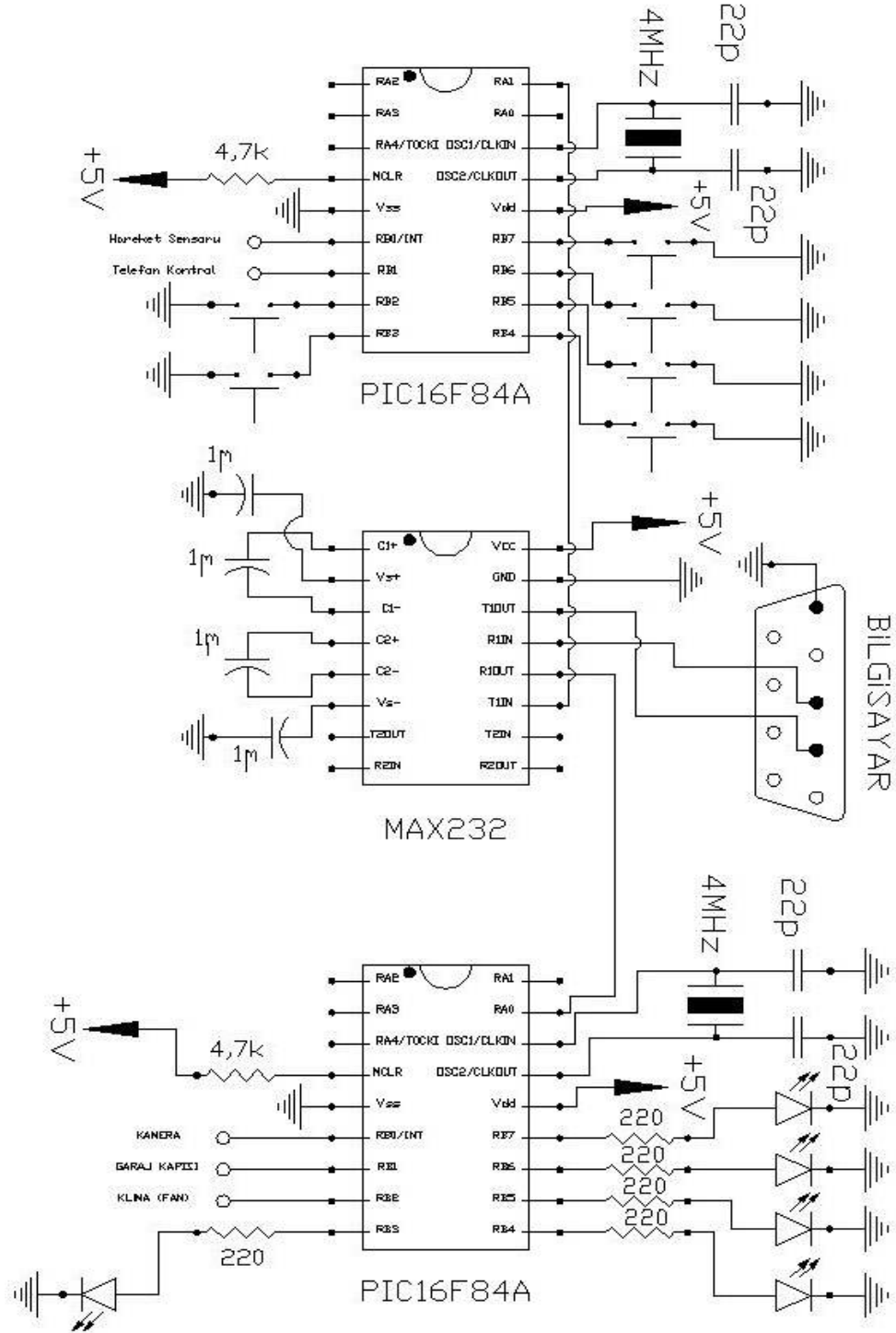
port_b_pullups(TRUE);

while (1)
{
    c = input_b();

    putc ( 'P' );
    for (i = 0; i < 8; i++)
    {
        s = (c & 0x80) >> 7 ;
        c = c << 1;
        s = s + '0';
        putc ( s );
    }
}
}
```

BÖLÜM 8: PROJE DONANIMI

Projenin devresi Şekil 8.1’de görüldüğü gibidir.



Şekil 8.1: Projenin Devresi

SONUÇ

Proje kapsamında 2 adet DC Servo motor, 1 adet klima modeli olarak kullanılan fan ve 5 adet LED sürülmüştür. Fakat çıkışlara sürme devreleri ile veya rölelerle bağlanacak herhangi bir elektrikli cihaz da kolaylıkla kontrol edilebilir.

Ayrıca 1 tanesi hareket sensöründen, 1 tanesi telefon hattı üzerinden ve 6 tanesi de pushbuttonlardan olmak toplam 8 adet geri besleme bilgisi elde edilmiştir.

Ayrıca projenin görselliğinin iyileştirilmesi açısından RF kamera gibi çevresel birimlerden de yararlanılmıştır.

Proje geliştirilirken hiçbir donanımsal değişikliğe mahal vermiyecek şekilde sadece yazılımı değiştirerek farklı uygulamalarda kullanılabilecek şekilde düşünülmüştür.

Tasarlanmış olan grafik kullanıcı arayüz daha da iyileştirilerek zaman tabloları oluşturularak belli tarih ve saatlerde istenilen çıkışların değiştirilmesi sağlanıp, bunlar arşivlenebilirdi. Ancak bu projenin zaman ve konusunun bir miktar dışına çıkmaktadır. Bu yüzden üzerinde minimum düzeyde durulmuştur.

Akıllı binalar günümüz teknolojisi sayesinde çok yaygınlaşmaktadırlar. Proje geliştirilirken bu tür yerlerde kullanılmasına uygun olduğu da göz önünde bulundurulmuştur.

KAYNAKLAR

PIC ve PC İletişim Projeleri, Prof. Dr. Dođan İBRAHİM, BİLEŞİM Yayınevi, Mart 2004, İstanbul

Mikrodenetleyiciler ve PIC Programlama, Orhan ALTINBAŞAK, ALTAŞ Yayınları, Ekim 2004, İstanbul

Anahtar Makinesinin Bilgisayar Destekli Kontrolü, Araş. Gör. Onur AKBATI, Yıldız Teknik Üniversitesi – Elektrik/Elektronik Fakültesi – Elektrik Mühendisliği Lisans Bitirme Tezi, 2005, İstanbul

Telefon Hattı ile Cihaz Kontrolü, Serdar ATEŞ, Yıldız Teknik Üniversitesi – Elektrik/Elektronik Fakültesi – Elektrik Mühendisliği Lisans Proje – II Çalışması, 2005, İstanbul

PIC16F84A User Manuel

MAX232 User Manuel

HT9170 User Manuel

ÖZGEÇMİŞ

Ad Soyad: Serdar ATEŞ
Doğum Tarihi : 24.12.1982
Doğum Yeri : Keşan – EDİRNE
Lise : 1997 – 2001, Keşan Yusuf Çapraz Lisesi
Staj Yaptığı Yerler : ARÇELİK A.Ş., Tekirdağ (6 Hafta)
PROFİLO Elektronik A.Ş., Tekirdağ (5 Hafta)