

**TİCARİ KURULUŞLARDA VE SOSYAL  
TESİSLERDE RFID (RADYO FREKANSI İLE  
TANIMLAMA) TEKNOLOJİSİ İLE  
OTOMASYON**

**MEHMET MÜTİN TÜRK**

**MERSİN ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ  
ANA BİLİM DALI**

**YÜKSEK LİSANS TEZİ**

**MERSİN  
ARALIK – 2012**

**TİCARİ KURULUŞLARDA ve SOSYAL  
TESİSLERDE RFID (RADYO FREKANSI İLE  
TANIMLAMA) TEKNOLOJİSİ İLE  
OTOMASYON**

**MEHMET MÜTİN TÜRK**

**MERSİN ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ  
ANA BİLİM DALI**

**YÜKSEK LİSANS TEZİ**

**Danışman  
Yrd. Doç. Dr. Ali YILDIZ**

**MERSİN  
ARALIK – 2012**

Mehmet Mütin TÜRK tarafından Yrd. Doç. Dr. Ali YILDIZ danışmanlığında hazırlanan “Ticari Kuruluşlarda ve Sosyal Tesislerde RFID (Radyo Frekansı ile Tanımlama) Teknolojisi ile Otomasyon” başlıklı bu çalışma aşağıda imzaları bulunan jüri üyeleri tarafından oy birliği ile Yüksek Lisans Tezi olarak kabul edilmiştir.

İmza

Prof. Dr. Caner ÖZDEMİR

Yrd. Doç. Dr. Ali YILDIZ

Yrd. Doç. Dr. Zeki YETGİN



Yukardaki Jüri kararı Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 28.../12.../2012...tarih ve 2012..24../.....799..... sayılı kararıyla onaylanmıştır.

  
Prof. Dr. A. Murat GİZİR  
Enstitü Müdürü



## **TİCARİ KURULUŞLARDA ve SOSYAL TESİSLERDE RFID (RADYO FREKANSI İLE TANIMLAMA) TEKNOLOJİSİ İLE OTOMASYON**

**Mehmet Mütin TÜRK**

### **ÖZ**

Günümüzde, RFID teknolojisinin otomasyon uygulamalarında kullanımı oldukça yaygındır. Programlanabilmeleri, farklı çalışma mesafelerine sahip olmaları, maliyetleri ve iş gücünü azaltmaları bunun en önemli nedenleridir. Kütüphaneler, ürün takibi ve sınıflandırılması, araç takip ve park sistemleri, sağlık sektörü ve otomatik geçiş sistemleri uygulama alanlarından sadece bir kaçıdır. Bu tez kapsamında bir ticari kuruluş olan oteller için bir örnek RFID sistemi geliştirildi. Bu sistem ile zaman ve personel maliyetlerinin azaltılması, işletmenin kaynaklarının düzgün bir şekilde dağıtılmasına yardımcı olunması hedeflendi. Sistem, üç RFID okuyucusundan, bir kaç RFID etiketinden, bir bilgisayardan ve mikrodenetleyici tarafından kontrol edilen bir kontrol devresinden oluşmaktadır. İki bilgisayar, biri mikrodenetleyici için olmak üzere üç adet yazılım geliştirildi. Bu yazılımlar sayesinde girişlerin kontrolü ve otomatik ödeme sistemi gerçekleştirildi. Ve otelde hangi bölümün hangi zaman aralığında daha fazla tercih edildiği tespit edildi. Böylece istenilen hedeflere önemli ölçüde ulaşılmış oldu.

**Anahtar Kelimeler:** RFID, Otomasyon

**Danışman:** Yrd. Doç. Dr. Ali YILDIZ, Elektrik-Elektronik Mühendisliği Ana Bilim Dalı, Mersin Üniversitesi

## **AUTOMATION WITH RFID (Radio Frequency Identification) TECHNOLOGY IN COMMERCIAL AND SOCIAL ORGANIZATIONS**

**Mehmet Mütin TÜRK**

### **ABSTRACT**

Nowadays, use of RFID technology is quite common in automation applications. Being able to be programmed, having varying operating distances, reducing labor and operating costs are the most important reasons. Libraries, product tracking and classification, vehicle tracking and park systems, health sector are just a few examples of its application areas. In this thesis, an example RFID system was built for an hotel, a commercial organization. With this system, the goals were helping the company to distribute their resources properly and to reduce time and personnel costs. The system consists of three RFID readers, a few RFID tags, a computer and an electronic circuit which is controlled by a microcontroller. In total, three softwares were developed, two for the computer and one for the microcontroller. With these softwares, entrance control and payments were done automatically. And at which time and day which places were preferred in the hotel was determined. Thus, the desired goals were mostly achieved.

**Key Words:** RFID, Automation

**Advisor:** Assist. Prof. Dr. Ali YILDIZ, Department of Electrical and Electronics Engineering, Mersin University

## **TEŞEKKÜR**

Yüksek lisans eğitimim boyunca desteğini esirgemeyen danışmanım sayın Yrd. Doç. Dr. Ali YILDIZ'a, çalışmamın çeşitli safhalarında bana yardımcı olan hocalarım sayın Prof. Dr. Caner Özdemir'e ve sayın Yrd. Doç. Dr. Zeki Yetgin'e teşekkür ederim.

## İÇİNDEKİLER

### Sayfa

|  |             |
|--|-------------|
| <b>ÖZ</b> .....                                  | <b>i</b>    |
| <b>ABSTRACT</b> .....                            | <b>ii</b>   |
| <b>TEŞEKKÜR</b> .....                            | <b>iii</b>  |
| <b>İÇİNDEKİLER</b> .....                         | <b>iv</b>   |
| <b>ÇİZELGELER DİZİNİ</b> .....                   | <b>vi</b>   |
| <b>ŞEKİLLER DİZİNİ</b> .....                     | <b>vii</b>  |
| <b>EKLER DİZİNİ</b> .....                        | <b>viii</b> |
| <b>SİMGE ve KISALTMALAR DİZİNİ</b> .....         | <b>ix</b>   |
| <br>   |             |
| <b>1. GİRİŞ</b> .....                            | <b>1</b>    |
| <br>   |             |
| <b>2. KAYNAK ARAŞTIRMALARI</b> .....             | <b>2</b>    |
| <br>   |             |
| 2.1. RFID ETİKETLERİ.....                        | 2           |
| 2.1.1. Pasif Etiketler .....                     | 2           |
| 2.1.1.1. Mikroçip.....                           | 3           |
| 2.1.1.2. Anten .....                             | 4           |
| 2.1.2. Aktif Etiketler.....                      | 4           |
| 2.1.3. Yarı-Aktif Etiketler .....                | 5           |
| 2.2. RFID OKUYUCULARI .....                      | 6           |
| 2.2.1. Verici.....                               | 6           |
| 2.2.2. Alıcı.....                                | 7           |
| 2.2.3. Mikroişlemci .....                        | 7           |
| 2.2.4. Bellek .....                              | 7           |
| 2.2.5. Giriş Çıkış Arayüzü .....                 | 7           |
| 2.2.6. Bağlantı Arayüzü .....                    | 7           |
| 2.2.7. Güç Birimi.....                           | 7           |
| 2.3. OKUYUCU İLE ETİKET ARASINDAKİ İLETİŞİM..... | 8           |
| 2.3.1. Modüleli Geri Saçılma .....               | 8           |
| 2.3.2. Verici Tipi .....                         | 8           |
| 2.3.3. Transponder Tipi.....                     | 8           |
| 2.4. RFID ÇALIŞMA FREKANSLARI .....              | 9           |
| 2.5. RFID TEKNOLOJİSİNİN AVANTAJLARI .....       | 10          |
| 2.5.1. Temas Gerektirmeme .....                  | 10          |
| 2.5.2. Veri Yazabilme .....                      | 10          |
| 2.5.3. Görüşten Bağımsızlık.....                 | 10          |
| 2.5.4. Farklı Okuma Mesafeleri .....             | 11          |

|   |           |
|---|-----------|
| 2.5.5. Farklı Veri Kapasiteleri.....  | 11        |
| 2.5.6. Çoklu Okuma Desteği.....   | 11        |
| 2.5.7. Sağlamlık .....  | 11        |
| 2.5.8. Akıllı Etiketler.....  | 11        |
| <br>  |           |
| 2.6. RFID TEKNOLOJİSİNİN LİMİTLERİ .....  | 12        |
| 2.6.1. Çevresel Faktörler .....   | 12        |
| 2.6.2. RF Enerjisinin Limitleri .....   | 12        |
| <br>  |           |
| 2.7. RFID TEKNOLOJİSİNİN DİĞER OTOMATİK TANIMLAMA<br>TEKNOLOJİLERİYLE KARŞILAŞTIRILMASI ..... | 13        |
| <br>  |           |
| <b>3. MATERYAL ve YÖNTEM.....</b>   | <b>14</b> |
| 3.1. RFID OKUYUCULAR.....   | 14        |
| 3.1.1. Okuyucu UART İletişimi.....  | 15        |
| 3.1.2. Select Tag Komutu.....   | 16        |
| 3.2. RFID ETİKETLER .....   | 17        |
| 3.3. YAZILIMLAR.....  | 18        |
| 3.3.1. RFID Konsol Uygulaması.....  | 19        |
| 3.3.2. RFID Masaüstü Uygulaması .....   | 22        |
| 3.3.3. PIC C Uygulaması .....   | 23        |
| <br>  |           |
| <b>4. BULGULAR ve TARTIŞMA.....</b>   | <b>26</b> |
| <br>  |           |
| <b>5. SONUÇLAR ve ÖNERİLER .....</b>  | <b>29</b> |
| <br>  |           |
| <b>KAYNAKLAR .....</b>  | <b>30</b> |
| <br>  |           |
| <b>EKLER.....</b>   | <b>31</b> |
| <br>  |           |
| <b>ÖZGEÇMİŞ.....</b>  | <b>62</b> |



## **ÇİZELGELER DİZİNİ**

### **Sayfa**

|   |    |
|---|----|
| Çizelge 3.1. Okuyucuya gönderilen bilgiler .....          | 15 |
| Çizelge 3.2. Okuyucudan alınan bilgiler .....             | 15 |
| Çizelge 3.3. Başarılı okuma halinde cevap baytları .....  | 17 |
| Çizelge 3.4. Başarısız okuma halinde cevap baytları ..... | 17 |

## ŞEKİLLER DİZİNİ

### Sayfa

|  |    |
|--|----|
| Şekil 2.1. Pasif bir etiketin iç yapısı.....                                     | 3  |
| Şekil 2.2. Mikroçipin temel bileşenleri .....                                    | 3  |
| Şekil 2.3. Aktif bir etiketin iç yapısı .....                                    | 5  |
| Şekil 2.4. Bir RFID okuyucunun iç yapısı .....                                   | 6  |
| Şekil 3.1. SonMicro SM132-USB RFID okuyucu.....                                  | 14 |
| Şekil 3.2. Sistemde kullanılan RFID etiketleri.....                              | 18 |
| Şekil 3.3. Bar iş parçaçığının algoritması .....                                 | 19 |
| Şekil 3.4. Havuz iş parçaçığının algoritması.....                                | 20 |
| Şekil 3.5. musteritable tablosunun şeması .....                                  | 21 |
| Şekil 3.6. bartable ve havuztable tablolarının şeması .....                      | 21 |
| Şekil 3.7. RFID Masaüstü Uygulaması bakiye güncelleme ekranı.....                | 22 |
| Şekil 3.8. Kart seçili iken RFID Masaüstü Uygulaması bakiye güncelleme ekranı .. | 23 |
| Şekil 3.9. Mikroelektronika USB UART kartı.....                                  | 24 |
| Şekil 3.10. Kontrol devresi .....  | 25 |
| Şekil 3.11. Kontrol devresi devre şeması.....                                    | 25 |
| Şekil 4.1. RFID Masaüstü Uygulaması müşteri kayıt ekranı.....                    | 26 |
| Şekil 4.2. Başarılı geçiş durumunda kontrol devresi.....                         | 27 |
| Şekil 4.3. RFID Masaüstü Uygulaması aylık istatistikler ekranı.....              | 28 |

## **EKLER DİZİNİ**

### **Sayfa**

|  |    |
|--|----|
| EK 1. mifare.cs Kaynak Kodları .....               | 31 |
| EK 2. RFID Konsol Uygulaması Kaynak Kodları.....   | 43 |
| EK 3. RFID Masaüstü Uygulaması Kaynak Kodları..... | 49 |
| EK 4. PIC C Uygulaması Kaynak Kodları .....        | 61 |

## **SİMGE ve KISALTMALAR DİZİNİ**

|      |   |
|------|---|
| RFID | Radio Frequency Identification, Radyo Frekansı ile Tanımlama                |
| OGS  | Otomatik Geçiş Sistemleri   |
| LF   | Low Frequency, Düşük Frekans  |
| HF   | High Frequency, Yüksek Frekans  |
| UHF  | Ultra High Frequency, Ultra Yüksek Frekans                                  |
| AC   | Alternating Current, Alternatif Akım  |
| DC   | Direct Current, Doğru Akım  |
| RF   | Radio Frequency, Radyo Frekansı   |
| OCR  | Optical Character Recognition, Optik Karakter Tanıma                        |
| USB  | Universal Serial Bus, Evrensel Seri Veriyolu                                |
| UART | Universal Asynchronous Receiver/Transmitter, Evrensel Asenkron Alıcı/Verici |
| TTL  | Transistor-Transistor Logic, Transistör-Transistör Mantığı                  |
| LED  | Light Emitting Diode, Işık Yayan Diyot                                      |
| bps  | Bytes Per Second, Saniyedeki Bayt Sayısı                                    |
| KB   | Kilobytes, Kilobayt   |

## **1. GİRİŞ**

RFID (radyo frekansı ile tanımlama) teknolojisinin kullanımı gün geçtikçe yaygınlaşmaktadır. Örneğin, Amerikan Demiryolları 1990 yılında vagon izlemede RFID teknolojisini kullanmaya başlamıştır [Üstündağ, 2008]. Amerika, 2003 yılında Irak Savaşı sırasında RFID teknolojisinden faydalanmıştır [Üstündağ, 2008]. Avrupa’da bir çok ülke OGS’lerde (otomatik geçiş sistemleri) RFID sistemlerini kullanmaktadır [Üstündağ, 2008]. Amerika kökenli Wal-Mart mağazalar zinciri ürünlerinin takibinde ve sınıflandırılmasında RFID teknolojisini kullanmaktadır [Glover and Bhatt, 2006]. RFID sistemlerinin, programlanabilmeleri, farklı çalışma mesafelerine sahip olmaları, maliyetleri ve iş gücünü azaltmaları bu yaygınlaşmanın en önemli nedenlerindedir. RFID sistemlerinin yakın gelecekte barkod ve akıllı kart gibi birçok otomatik tanımlama teknolojisinin yerini alması ön görülmektedir.

Bu tez ile ticari işletmeler için örnek bir RFID sistemi geliştirildi. Bilindiği üzere, günümüzde, ticari kuruluşlarda müşterilere verilen hizmetler ve ürünler için sürekli kayıt tutulması gerekmektedir. Müşteri herhangi bir hizmet veya ürün satın aldığı anda, bu hizmetin veya ürünün ücreti bir personel tarafından faturasına eklenmektedir. Veya müşteri kuruluştaki bir aktiviteye katılmak istediğinde, o aktivite ile ilgili herhangi bir sınırlama (yaş, cinsiyet vb.) varsa, personeller tarafından gerekli kontrollerin yapılması gerekmektedir. Bunun gibi durumlar kuruluşa ekstra hizmet ve zaman maliyeti getirmektedir. Bu tez ile bahsi geçen durumlara çözüm olarak örnek bir RFID sistemi geliştirildi. Kuruluş olarak bir otel ele alındı ve bu otelde iki sanal bölüm seçildi. Seçilen bölümlerin girişlerinde, geliştirilen bir yazılım aracılığıyla, gerekli tüm kontroller yapıldı ve bir otomatik ödeme sistemi gerçekleştirildi. Bunların yanısıra bölümlere yapılan tüm girişlerin tarihleri veritabanına kaydedildi.

Otomatik kontrol ve ödeme sistemi sayesinde, personellerin fatura kaydı alması ve girişlerde kontrol yapması gerekliliği ortadan kaldırıldı. Veritabanına kaydedilen tarihler sayesinde ise kuruluştaki hangi bölümün hangi zamanlarda daha fazla tercih edildiği belirlendi. Bu bilgiler sayesinde, kuruluşun kaynaklarını günlere ve saatlere göre daha düzgün dağıtması beklenmektedir.

## **2. KAYNAK ARAŞTIRMALARI**

RFID sistemleri aşağıdaki elemanlardan oluşmaktadır:

- RFID Etiketleri
- RFID Okuyucuları
- Kontrol Birimleri (Bilgisayarlar, mikrodenetleyiciler vb.)

### **2.1. RFID ETİKETLERİ**

RFID etiketi, veri depolayabilen ve RFID okuyucuya radyo dalgaları vasıtasıyla temassız olarak veri gönderebilen bir cihazdır. RFID etiketleri üçe ayrılmaktadır:

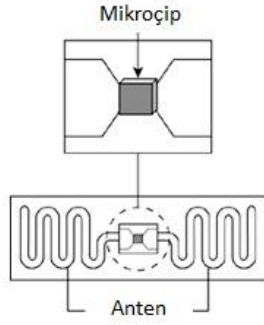
- Pasif Etiketler
- Aktif Etiketler
- Yarı-aktif (Yarı-pasif) Etiketler

#### **2.1.1. Pasif Etiketler**

Pasif etiketlerin üzerinde güç kaynağı bulunmaz. RFID okuyucudan gönderilen AC güç sinyalini kullanarak çalışırlar. Bu yüzden, pasif etiketlerin kullanıldığı sistemlerde, ilk önce okuyucu temasa geçer, daha sonra etiket cevap verir.

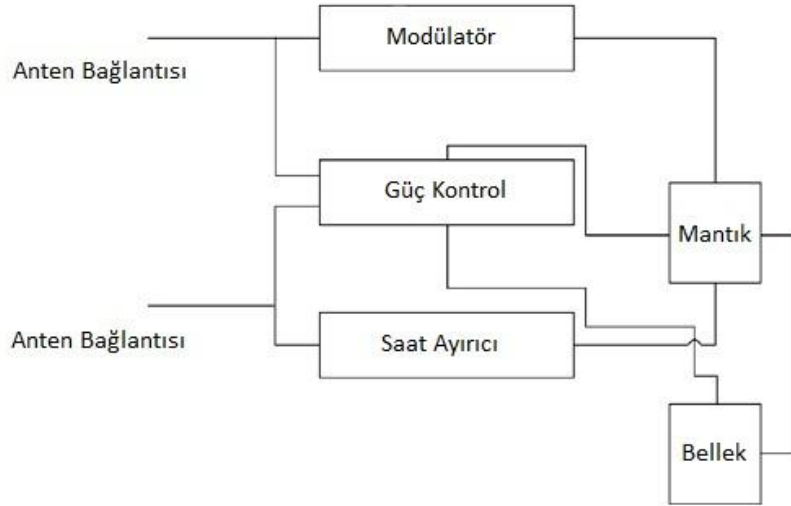
Uzun ömürlüdürler ve zorlu koşullara dayanabilirler. Örneğin, bazı pasif etiketler, asit gibi aşındırıcı maddelere, 200 °C gibi sıcaklıklara bile dayanabilmektedir [Lahiri, 2005]. Aktif ve yarı-aktif etiketlere oranla daha ucuzdurlar. Lâkin, daha kısa okuma mesafelerine sahiptirler. Okuma mesafeleri bir kaç santimetreden başlar, yaklaşık 9 metreye kadar çıkar [Lahiri, 2005]. Pasif Etiketler iki kısımdan oluşur:

- Anten
- Mikroçip



Şekil 2.1. Pasif bir etiketin iç yapısı

#### 2.1.1.1. Mikroçip



Şekil 2.2. Mikroçipin temel bileşenleri

Güç kontrol ünitesi, okuyucunun anteninden alınan sinyalin AC (alternatif akım) bileşenini DC (doğru akım) sinyale çevirerek, mikroçipin diğer ünitelerine çalışmaları için gerekli olan gücü sağlar. Saat ayırıcı ünitesi ise, okuyucu anteninden alınan sinyalden saat sinyalini ayırır. Modülâtör ünitesi alınan sinyali modüle eder. Etiket cevapı, modüle edilen bu sinyalin içine gömülür. Bu sinyal daha sonra etiket anteni vasıtasıyla okuyucuya gönderilir. Mantık ünitesi okuyucu ile etiket arasındaki iletişim protokolünü düzenlemekten sorumludur. Bellek birimi ise veri depolamak için kullanılır. Günümüzde mikroçipler bir kum tanesi büyüklüğünde

üretilebilmektedir. Fakat etiketlerin boyutunu belirleyen mikroçipler değil, antenlerdir [Lahiri, 2005].

#### 2.1.1.2. Anten

Etiket anteni, mikroçipe fiziksel olarak bağlıdır ve okuyucu anteninden gönderilen sinyalleri almak için kullanılır. Etiketler için sonsuz varyasyonda anten tasarımı mümkündür [Lahiri, 2005]. Antenin uzunluğu etiketin çalıştığı dalga boyu ile doğru orantılıdır.

#### 2.1.2. Aktif Etiketler

Aktif etiketlerin üzerinde, mikroçip ve antene ek olarak, entegre güç kaynağı (elektrik pili, güneş pili vb.) ve belirli görevleri yerine getirmek için kullanılan entegre devre elemanları bulunmaktadır. Aktif etiketler, okuyucuya veri transfer etmek için gerekli olan gücü üzerlerindeki güç kaynağından elde ederler. Dolayısıyla, pasif etiketlerin aksine, veri transferi için okuyucudan güç çekmelerine gerek yoktur. Entegre devre elemanları; mikroişlemciler, sensörler, giriş/çıkış portları gibi devre elemanlarıdır ve dahili güç kaynağı tarafından enerjilendirirler. Bu elemanlar, ortamın sıcaklığını, basıncını, nem oranını ölçmek veya etiketin iliştiirildiği ürünün son tüketim tarihini belleğinde tutmak gibi değişik işlemler için kullanılabilirler. İstendiği takdirde bu bilgileri okuyucuya da gönderebilirler.

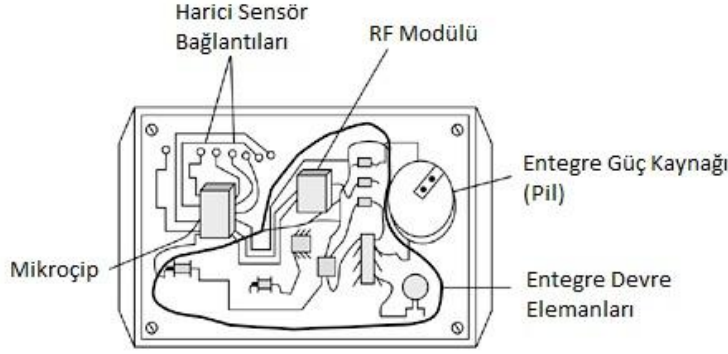
Aktif etiketler, buldukları ortama sürekli olarak ya da belirli aralıklarla sinyal gönderirler. Bu yüzden, aktif etiketlerin kullanıldığı sistemlerde, ilk önce etiket temasa geçer, daha sonra okuyucu cevap verir.

İki tip aktif etiket mevcuttur. Okuyucunun olmadığı durumlarda bile ortama veri gönderen aktif etiketlere “verici tipi aktif etiket” adı verilmektedir. Diğer tip aktif etiketler ise “transponder tipi aktif etiketler” dir. Bu tipteki aktif etiketler, okuyucunun olmadığı durumlarda bir tür uyku modunda ya da düşük güç modunda çalışırlar. Ve veri gönderimi yerine, ortamda okuyucu olup olmadığını sorgulamak için belli aralıklarla ortama bir sinyal gönderirler. Okuyucu bu sinyali aldığı zaman, etikete uyku modundan çıkması için bir uyandırma sinyali gönderir. Etiket,



okuyucunun gönderdiği bu uyandırma sinyalini aldığı zaman transponder tipi olarak çalışmayı bırakıp, verici tipi olarak çalışmaya başlar.

Bir aktif etiketin okuma mesafesi 30 metreden daha uzun olabilmektedir [Lahiri, 2005].

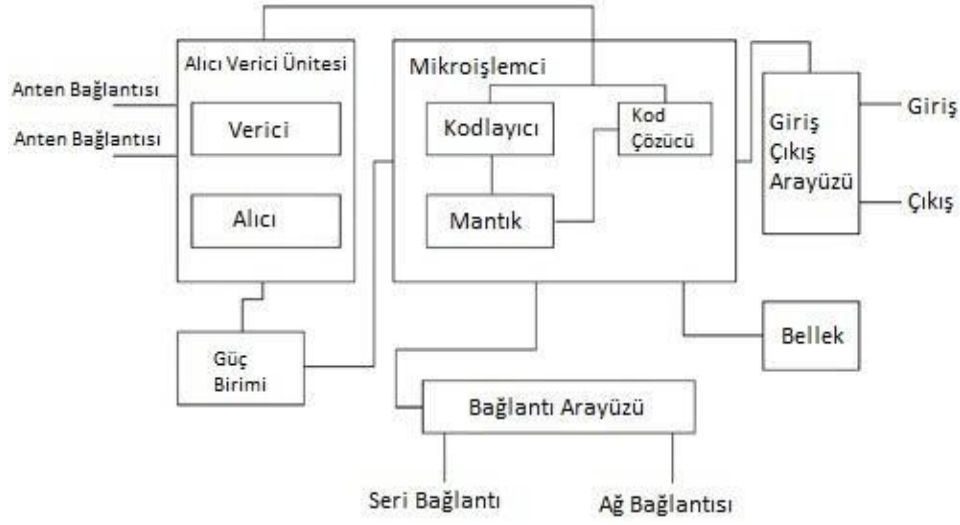


Şekil 2.3. Aktif bir etiketin iç yapısı

### 2.1.3. Yarı-Aktif (Yarı-Pasif) Etiketler

Yarı-aktif etiketlerin üzerinde, aktif etiketlerde olduğu gibi, entegre güç kaynağı ve entegre devre elemanları bulunmaktadır. Güç kaynağı etiketin devre elemanlarını çalıştırmak için gerekli olan gücü sağlar. Veri iletimi için gerekli olan güç ise güç kaynağından değil, okuyucudan çekilen güç ile sağlanır.

## 2.2. RFID OKUYUCULARI



Şekil 2.4. Bir RFID okuyucunun iç yapısı

RFID okuyucuları, desteklediği RFID etiketlerindeki verileri okuyabilen ve bu etiketlere veri yazabilen cihazlardır. Bir RFID okuyucusu aşağıdaki elemanlardan oluşur:

- Verici
- Alıcı
- Mikroişlemci
- Bellek
- Giriş Çıkış Arayüzü
- Bağlantı Arayüzü
- Güç Birimi

### 2.2.1. Verici

Okuyucunun etikete AC güç sinyali ile saat sinyalini göndermesinden sorumlu birimdir.

### 2.2.2. Alıcı

Etiketden gelen analog sinyalleri okuyucu anteni vasıtasıyla alan birimdir. Daha sonra bu analog sinyali sayısal bir sinyale çevirmesi için mikroişlemciye gönderir.

### 2.2.3. Mikroişlemci

Alıcıdan gönderilen analog sinyali sayısal sinyale çevirir. Ayrıca, etiket ile okuyucu arasındaki iletişim protokolünü düzenlemekten sorumludur.

### 2.2.4. Bellek

Yapılan etiket okumalarının bir listesi ve okuyucu konfigürasyon parametreleri gibi verileri hafızasında tutabilen birimdir. Böylece, okuyucu ile kontrol ünitesi arasındaki bağlantı koptuğunda bile bazı bilgiler korunabilmektedir.

### 2.2.5. Giriş Çıkış Arayüzü

Sensörler, eyleyiciler ve uyarıcılar için bağlantı sağlar. Seçeneğe bağlıdır, her okuyucuda bulunmaz.

### 2.2.6. Bağlantı Arayüzü

Bilgisayarlar ve mikrodenetleyiciler gibi kontrol ünitelerine bağlantı sağlar. Bu bağlantılar seri bağlantılar veya ağ bağlantıları olabilmektedir.

### 2.2.7. Güç Birimi

Genellikle elektrik şebekesine bağlanan bir güç kablosuyla okuyucuya çalışması için gerekli olan gücü sağlar.

## 2.3. OKUYUCU İLE ETİKET ARASINDAKİ İLETİŞİM

Okuyucu ile etiket arasındaki iletişim üç farklı şekilde olabilmektedir:

- Modüleli Geri Saçılma
- Verici Tipi
- Transponder Tipi

### 2.3.1. Modüleli Geri Saçılma

Bu tipteki iletişim hem pasif hem de yarı-pasif etiketler için geçerlidir. Okuyucu etikete AC güç sinyali ve saat sinyali içeren bir sürekli radyo dalgası gönderir. Etiket anteni AC güç sinyalini DC güç sinyaline çevirerek, mikroçipe güç sağlar. Alınan sinyalden saat sinyali ayrıştırılır. Etiket, alınan sinyali, içine kendi verisini de gömerek modüle eder ve okuyucu antenine geri gönderir. Okuyucu alınan bu sinyali çözer ve etiket verisini okur. Mikroçip genellikle okuma işlemleri için yaklaşık 1.2 volta, yazma işlemleri için de yaklaşık 2.2 volta ihtiyaç duyar [Lahiri, 2005].

### 2.3.2. Verici Tipi

Bu tipteki iletişim sadece aktif etiketler için geçerlidir. Etiket anteni, etiket verisini belli aralıklarla ortama gönderir. Bu gönderme işlemi ortamda okuyucu olsa da olmasa da gerçekleşir.

### 2.3.3. Transponder Tipi

Bu iletişim tipi transponder tipi aktif etiketler için geçerlidir. Etiket, okuyucunun olmadığı durumlarda bir tür uyku modunda ya da düşük güç modunda çalışır. Ve veri gönderimi yerine, ortamda okuyucu olup olmadığını sorgulamak için belli aralıklarla ortama bir sinyal gönderir. Okuyucu bu sinyali aldığı zaman, etikete uyku modundan çıkması için bir uyandırma sinyali gönderir. Etiket, okuyucunun gönderdiği bu uyandırma sinyalini aldığı zaman transponder tipi olarak çalışmaya bırakıp, verici tipi olarak çalışmaya başlar.

## 2.4. RFID ÇALIŞMA FREKANSLARI

RFID sistemleri dört farklı frekans bandında çalışırlar. Bunlar:

- LF (125-134 Khz) (Düşük frekans)
- HF (13.56 MHz) (Yüksek frekans)
- UHF (860-960 MHz) (Ultra yüksek frekans)
- Mikrodalga (2.45 GHz ve 5.8 GHz)

Kullanılacak olan frekans bandı tamamen uygulamaya bağlıdır. Uzun okuma mesafesi (10 metreden uzun) isteniyorsa UHF ya da mikrodalga frekans bandında çalışan sistemler kullanılması gerekecektir. Aynı zamanda, sistemin kullanılacağı ortam da frekans seçimi için önemlidir. Örneğin, bazı nesnelere belli bir frekans aralığında radyo dalgalarını geçirirken, diğer frekanslarda yansıtılmakta ya da emebilmektedir.

## 2.5. RFID TEKNOLOJİSİNİN AVANTAJLARI

RFID teknolojisinin avantajlarını aşağıdaki gibi sıralayabiliriz:

- Temas Gerektirmeme
- Veri Yazabilme
- Görüşten Bağımsızlık
- Farklı Okuma Mesafeleri
- Farklı Veri Kapasiteleri
- Çoklu Okuma Desteği
- Sağlamlık
- Akıllı Etiketler

### 2.5.1. Temas Gerektirmeme

Okuma ve yazma işlemleri için etiketlerle okuyucular arasında fiziksel temas şart değildir.

### 2.5.2. Veri Yazabilme

Yeniden yazılabilir belleğe sahip etiketlere yaklaşık 100.000 defaya kadar tekrar tekrar veri yazılabilmektedir [Lahiri, 2005].

### 2.5.3. Görüşten Bağımsızlık

Okuma ve yazma işlemleri için okuyucunun etiketi birebir görmesine gerek yoktur. Bu özellik ürün takibinde ve diğer benzer uygulamalarda önemli bir avantaj sağlar. Örnek olarak, bir karton kutunun içine yerleştirilmiş bir ürüne iliştilen bir etiket sayesinde, kutuyu açmadan ürünün ne olduğu, ne gibi özelliklere sahip olduğu öğrenilebilir.

#### 2.5.4. Farklı Okuma Mesafeleri

Etiket okuma mesafeleri bir kaç santimetreden başlayıp yaklaşık 90 metreye kadar çıkabilmektedir [Lahiri, 2005]. Düşük frekans bandında çalışan bir pasif etiket okunma mesafesi birkaç santimetre iken, UHF bandında çalışan bir aktif etiket okunma mesafesi onlarca metreyi bulabilmektedir.

#### 2.5.5. Farklı Veri Kapasiteleri

Teorik olarak, etiketler istenilen her veri kapasitesinde üretilebilirler [Lahiri, 2005].

#### 2.5.6. Çoklu Okuma Desteği

RFID okuyucuları aynı anda sadece bir tane etiket okuyabilmektedir. Fakat kullanılan farklı çakışma (anti-collision) algoritmaları sayesinde, çok kısa bir süre içinde birden fazla etiket okumak mümkündür.

#### 2.5.7. Sağlamlık

Pasif etiketler; ısı, nem, aşındırıcı kimyasal maddeler, mekanik titreşim ve şok gibi çevresel faktörlere karşı dayanmak üzere üretilebilirler. Örneğin, piyasadaki bazı pasif etiketler 200 °C gibi sıcaklıklara bile dayanabilmektedir [Lahiri, 2005]. Günümüzde tüm çevresel faktörlerin hepsine birden dayanacak tek bir etiket üretmek mümkün değildir.

#### 2.5.8. Akıllı Etiketler

Aktif ve yarı-aktif etiketler ortamdaki ve sistemdeki değişkenleri (ortam basıncı, ortam sıcaklığı vb.) ölçmek ve görüntülemek üzere tasarlanabilirler.

## 2.6. RFID TEKNOLOJİSİNİN LİMİTLERİ

RFID sistemleri için aşağıdaki limitlerden söz etmek mümkündür:

- Çevresel Faktörler
- RF Enerjisinin Limitleri

### 2.6.1. Çevresel Faktörler

RFID sisteminin kurulu olduğu ortamda yüksek oranda sıvı, metal gibi maddelerin varlığı belirli frekanslarda sistemin çalışmasını olumsuz etkileyebilmektedir. Örneğin, yüksek UHF bandını kullanan bir RFID sisteminde, etiket ile okuyucu arasına bir insan girerse, okuma gerçekleşmeyecektir. Bunun nedeni insan vücudunun büyük bir kısmının sudan oluşuyor olmasıdır ve su, yüksek UHF bandında radyo dalgalarını emer. Aynı zamanda, ortamda aynı frekans bandını kullanan elektronik cihazlar varsa RFID sistemi oluşan gürültüden olumsuz etkilenecektir.

### 2.6.2. RF Enerjisinin Limitleri

RF (radyo frekansı) enerjisinin delip - geçme gücünün (penetrating power) bir sınırı vardır. Örneğin, tek bir kağıdın arkasına konan bir etiket okunabilirken, yüz kağıdın arkasına konan bir etiket okunamayabilir. Okuyucunun ortama yayabileceği elektromanyetik dalganın efektif gücü belirli düzenlemelerle sınırlandırılmış durumdadır. Bu düzenlemeler ülkeden ülkeye farklılık göstermektedir. Bu sorunun çözülebilmesi için var olan düzenlemelerin değiştirilmesi gerekmektedir.



## 2.7. RFID TEKNOLOJİSİNİN DİĞER OTOMATİK TANIMLAMA TEKNOLOJİLERİYLE KARŞILAŞTIRMASI

Literatürde bir çok farklı otomatik tanımlama teknolojisi bulunmaktadır. Bunlar; barkod, OCR (optik karakter tanıma), biyometrik tanıma, ses tanıma, akıllı kart gibi teknolojilerdir.

OCR, barkod ve ses tanıma teknolojilerinde kopyalama mümkündür. Bu da güvenlik açıklarına neden olabilmektedir. RFID, biyometrik tanıma ve akıllı kart teknolojilerinde ise kopyalama mümkün değildir [Müller, 2010].

Biyometrik tanıma teknolojisi çok pahalı bir teknoloji olduğundan günlük hayatta kullanımı yüksek maliyet gerektirmektedir. Yapılan okumalar, temaslı (parmak izi tanımlama) ya da kısa mesafede (retina tanımlama) gerçekleşmektedir.

Akıllı kart teknolojisi bahsi geçen teknolojiler içinde RFID teknolojisine en yakın teknolojidir. Fakat akıllı kartlar, RFID teknolojisinin aksine; nem, kir gibi etmenlerden olumsuz yönde etkilenmekte ve okuyucuya direk temas gerektirmektedir [Müller, 2010].

RFID teknolojisi bahsi geçen negatif unsurlardan hiç birini barındırmaz. Etiketler; temassızdır (metrelerce uzaktan okunabilir), kopyalanamaz, programlanabilir ve hava şartlarından etkilenmez.

### 3. MATERYAL ve YÖNTEM

Sistemde donanım olarak, üç adet RFID okuyucusu, beş adet RFID etiketi, bir bilgisayar ve mikrodenetleyici kontrollü bir kontrol devresi kullanılmıştır. Kontrol işlemleri geliştirilen üç uygulama tarafından sağlanmaktadır. Bu uygulamalardan iki tanesi bilgisayarda, bir tanesi ise mikrodenetleyicide çalışmaktadır.

#### 3.1. RFID OKUYUCULAR



Şekil 3.1. SonMicro SM132-USB RFID okuyucu

Sistemde SonMicro firmasının üretmiş olduğu, 13.56 MHz frekansında çalışan, SM132-USB model numarasına sahip üç adet RFID okuyucu kullanılmıştır. Okuyucular ile bilgisayar arasındaki iletişim USB (evrensel seri veriyolu) portu aracılığıyla gerçekleşmektedir. Okuyucuların çalışması için gerekli olan güç USB portundan sağlanmaktadır.

Okuyucularla beraber, yazılım geliştirmeyi kolaylaştırmak amacıyla, mifare.cs olarak adlandırılan, bir C# kütüphanesi verilmektedir. Bu kütüphanenin kaynak kodları SonMicro firmasının izniyle EK 1'de verilmiştir. Kütüphane, okuyucuda tanımlanmış komutların C# dilindeki metodlara dönüştürülmüş halidir. Oluşturulan sistemde bu metodlardan sadece CMD\_SelectTag() metodu kullanılmıştır. CMD\_SelectTag() metodu okuyucudaki Select Tag komutunun C#

diline uyarlanmış halidir. Select Tag komutu kullanıldığında okuyucunun RF alanında bir etiket mevcutsa, okuyucudan cevap olarak o etiketin seri numarası alınır.

### 3.1.1. Okuyucu UART İletişimi

Okuyucu ile bilgisayar arasındaki iletişim seri iletişim şeklinde gerçekleşmektedir. USB – seri UART (evrensel asenkron alıcı/verici) dönüşümü okuyucu üzerindeki FTDI firmasının ürettiği FT232RL çipi ile sağlanmaktadır. Okuyucu seri porttan 19200 bps (bir saniyedeki bayt sayısı), 38400 bps, 57600 bps ve 115200 bps hızlarında iletişim gerçekleştirebilmektedir. Ayarlarda değişiklik yapılmazsa okuyucu 19200 bps hızında çalışmaktadır. İletişimde 8-N-1 konfigürasyonu kullanılmaktadır. Bu konfigürasyonda veri bit sayısı 8'dir, eşlik biti yoktur ve durma biti 1'dir.

Okuyucuya gönderilen ve okuyucudan alınan bilgiler aşağıdaki çizelgelerde verilmiştir:

Çizelge 3.1. Okuyucuya gönderilen bilgiler

|         | Başlık Baytı | Rezerve Bayt | Uzunluk Baytı | Komut Baytı | Veri Baytları | Kontrol Baytı |
|---------|--------------|--------------|---------------|-------------|---------------|---------------|
| Uzunluk | 1 bayt       | 1 bayt       | 1 bayt        | 1 bayt      | Değişken      | 1 bayt        |

Çizelge 3.2. Okuyucudan alınan bilgiler

|         | Başlık Baytı | Rezerve Bayt | Uzunluk Baytı | Komut Baytı | Cevap Baytları | Kontrol Baytı |
|---------|--------------|--------------|---------------|-------------|----------------|---------------|
| Uzunluk | 1 bayt       | 1 bayt       | 1 bayt        | 1 bayt      | Değişken       | 1 bayt        |

Başlık Baytı: Veri transferinde kullanılan başlangıç baytıdır. Değeri daima 0xFF'dir.

Rezerve Bayt: Üretici firmanın okuyucuya eklediği fakat şu an için kullanılmayan bir bayttır. Değeri daima 0x00'dir.

Uzunluk Baytı: Komut ve veri/cevap baytlarının toplam kaç bayttan oluştuğunu belirtir. 1 bayt uzunluğundadır.

Komut Baytı: Bu bayt okuyucunun gerçekleştireceği işlemi belirtmektedir. Örneğin, Select Tag komutu için bu değer 0x83'tür.

Veri Baytları: Veri baytları değişken uzunluktadırlar ve komutun gerçekleşmesi için ihtiyaç duyulan verilerdir. Select Tag komutunda herhangi bir veri kullanılmamaktadır. Etikete veri yazma işlemlerinde ise bu kısım etikete yazılacak olan veridir.

Cevap Baytları: Okuyucudan alınan bilgilerdir. Değişken uzunluktadırlar.

Kontrol Baytı: Başlık baytı dışındaki tüm baytların toplamıdır. İşlemde hata olup olmadığını kontrol etmek için kullanılmaktadır.

### 3.1.2. Select Tag Komutu

Select Tag komutu kullanıldığı anda okuyucunun RF alanında bir etiket mevcutsa okuyucudan cevap olarak o etiketin seri numarası alınır. Okuyucu, çok kısa bir süre içinde birden fazla etiket okuyabilmek için, Select Tag komutu ile aynı anda bir çakışma (anti-collision) algoritması da çalıştırmaktadır.

Select Tag komutu için veri baytları bulunmamaktadır. Komut baytı ise 0x83'tür. Cevap baytları okuma başarılı ise 6 veya 9 bayt, okuma başarısız ise 2 bayt uzunluğundadır. Okumanın başarılı olduğu durumlarda, kullanılan etiket Mifare 1K veya Mifare 4K ise cevap baytları 6 bayt, kullanılan etiket Mifare Ultralight ise cevap baytları 9 bayt uzunluğundadır.

Çizelge 3.3. Başarılı okuma halinde cevap baytları

|                        |   |
|------------------------|---|
| Uzunluk                | 6 veya 9 bayt   |
| Komut                  | 1 bayt<br>0x83  |
| Etiket tipi            | 1 bayt<br>0x01 Mifare Ultralight<br>0x02 Mifare Standart 1K<br>0x03 Mifare Classic 4K<br>0xFF Bilinmeyen Etiket |
| Etiketin seri numarası | 4 veya 7 bayt   |

Çizelge 3.4. Başarısız okuma halinde cevap baytları

|           |  |
|-----------|--|
| Uzunluk   | 2 bayt   |
| Komut     | 1 bayt<br>0x83   |
| Hata Kodu | 1 bayt<br>0x4E Etiket bulunamadı<br>0x55 Okuyucunun RF alanı inaktif |

### 3.2. RFID ETİKETLER

Sistemde beş adet RFID etiketi kullanılmıştır. Bu etiketlerden dört tanesi kart şeklinde, bir tanesi ise anahtarlık şeklindedir. Kart şeklinde olanların model numarası MIFARE 1K-ISO, anahtarlık şeklinde olanın model numarası ise MIFARE 1K-KEY'dir. Her iki model de, şekildeki farklar dışında aynı kartlardır ve MIFARE Classic 1K olarak bilinmektedir. Bu kartlar 1 KB (kilobayt) yazılabilir belleğe

sahiptirler. Oluşturulan sistemde bu belleğe herhangi bir veri girilmemiştir. Bunun yerine, daha güvenli olacağı düşünüldüğünden yerel bir veritabanı tercih edilmiştir.



Şekil 3.2. Sistemde kullanılan RFID etiketleri

### 3.3. YAZILIMLAR

Microchip PIC 16F877A mikrodenetleyicisi üzerinde çalışması için bir, Microsoft Windows işletim sistemleri üzerinde çalışması için iki uygulama geliştirilmiştir. PIC uygulaması C dili ile yazılmış olup CCS C Compiler ile derlenmiştir. Windows uygulamaları ise Microsoft Visual Studio yazılımı, Microsoft SQL Server Compact Edition veritabanı ve C# programlama dili kullanılarak yazılmıştır.

RFID Konsol Uygulaması olarak adlandırılan birinci Windows uygulaması, işletim sistemi ile birlikte açılmakta ve arka planda sürekli olarak çalışmaktadır. Bu uygulama otelde sanal olarak seçilen iki yeri kontrol etmektedir. Bu sanal yerler bar ve havuz olarak seçilmiştir. Bu iki yerde birer adet RFID okuyucusu bulunmaktadır.

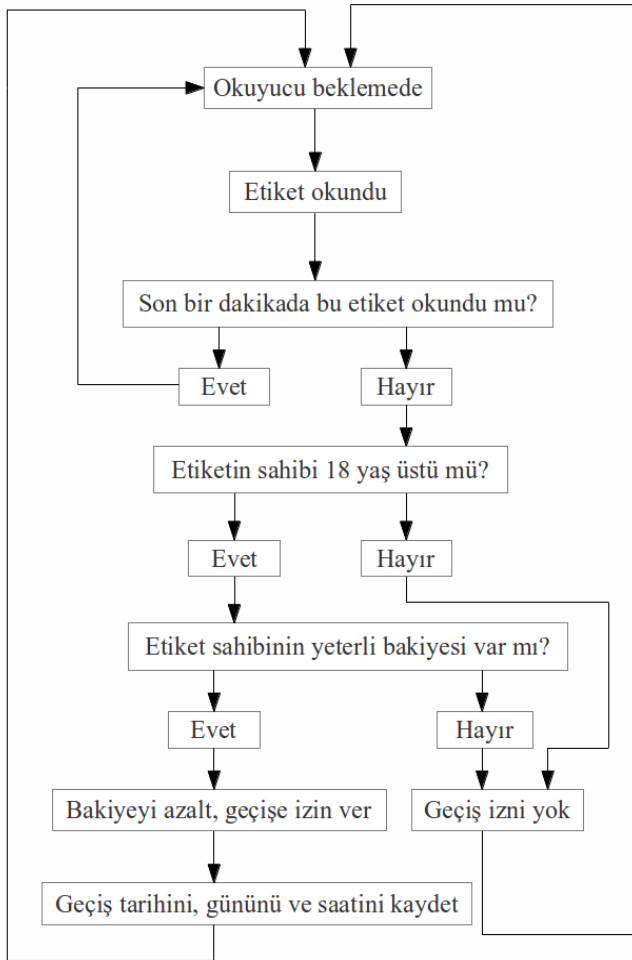
İkinci uygulama ise kullanıcı arayüzüne sahip bir masaüstü uygulamasıdır. RFID Masaüstü Uygulaması olarak adlandırılan bu uygulama işletmedeki kasada çalışması için tasarlanmıştır. Kayıt ekleme, kayıt yenileme, bakiye güncelleme, kayıp kart ve müşteri çıkış işlemlerinden sorumludur. Bar ve havuzdan yapılan girişlerin

sayısına ve bu girişlerin gün ve saat dağılımlarına da yine bu uygulama aracılığıyla erişilmektedir.

PIC 16F877A mikrodenetleyicisi bilgisayar ile USB portu üzerinden haberleşmektedir. Mikrodenetleyici, RFID Konsol Uygulaması'ndan aldığı veriler doğrultusunda ya yeşil bir LED'i (ışık yayan diyot) ya da kırmızı bir LED'i yakmaktadır.

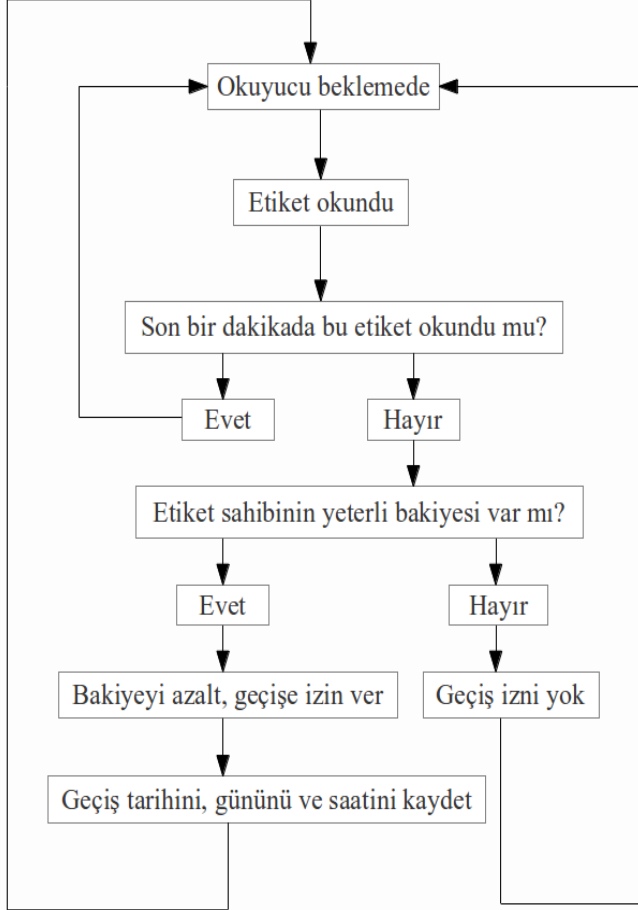
### 3.3.1. RFID Konsol Uygulaması

Uygulamada üç tane iş parçacığı (thread) bulunmaktadır. Birinci iş parçacığı havuzu, ikinci iş parçacığı ise barı kontrol etmektedir. Üçüncü iş parçacığı ise 24 saatte bir, veritabanından, bir aydan eski olan tarih kayıtlarını silmektedir. Havuz ve bar iş parçacıklarının algoritmaları aşağıdaki şekillerde verilmiştir:



Şekil 3.3. Bar iş parçacığının algoritması

Bar girişinde, yaşı onsekizden büyük olan müşterinin, yeterli bakiyesi olması durumunda girişine izin verilmektedir. Giriş gerçekleştiği anda, müşterinin bakiyesi düşürülmekte ve giriş tarihi, günü, saati veritabanına eklenmektedir.



Şekil 3.4. Havuz iş parçaçığının algoritması

Havuz girişinde yaş kontrolü yapılmamakta fakat barda olduğu gibi bir otomatik ödeme sistemi uygulanmakta ve giriş tarihleri veritabanına kaydedilmektedir.

Bar ve havuz girişleri, kontrol devresi aracılığıyla simüle edilmiştir. Devrede bulunan LED'ler bölüm girişlerindeki kapıları temsil etmektedir. Kırmızı LED'in yakılması geçiş izninin olmadığı, yani kapının açılmadığı durumu; yeşil LED'in yakılması ise geçiş izninin olduğu, yani kapının açıldığı durumu göstermektedir.



Veritabanında üç adet tablo bulunmaktadır. *bartable* ve *havuztable* olarak adlandırılan tablolar, geçişlerin olduğu tarihlerin, saatlerin ve günlerin kayıtlarını tutmaktadır. Bardan yapılan geçişlerin kayıtları *bartable* tablosunda, havuzdan yapılan geçişlerin kayıtları ise *havuztable* tablosunda tutulmaktadır. *musteritable* olarak adlandırılan üçüncü tablo ise müşterilere ait bilgilerin kayıtlarını tutmaktadır.

*bartable* ve *havuztable* tabloları aynı şemaya sahiptirler fakat içerdikleri bilgiler farklıdır. Veritabanındaki tabloların şemaları aşağıdaki şekillerde verilmiştir:

| Column Name | Data Type | Length | Allow Nulls | Unique | Primary Key |
|-------------|-----------|--------|-------------|--------|-------------|
| bakiye      | int       | 4      | Yes         | No     | No          |
| tagno       | nvarchar  | 100    | No          | No     | No          |
| tcno        | nvarchar  | 100    | No          | Yes    | No          |
| dogumyili   | int       | 4      | No          | No     | No          |
| adsoyad     | nvarchar  | 100    | No          | No     | No          |

Şekil 3.5. *musteritable* tablosunun şeması

| Column Name | Data Type | Length | Allow Nulls | Unique | Primary Key |
|-------------|-----------|--------|-------------|--------|-------------|
| date        | datetime  | 8      | No          | No     | No          |
| gun         | nvarchar  | 100    | No          | No     | No          |
| saat        | nvarchar  | 100    | No          | No     | No          |

Şekil 3.6. *bartable* ve *havuztable* tablolarının şeması

RFID Konsol Uygulaması'nın kaynak kodları EK 2'de verilmiştir.

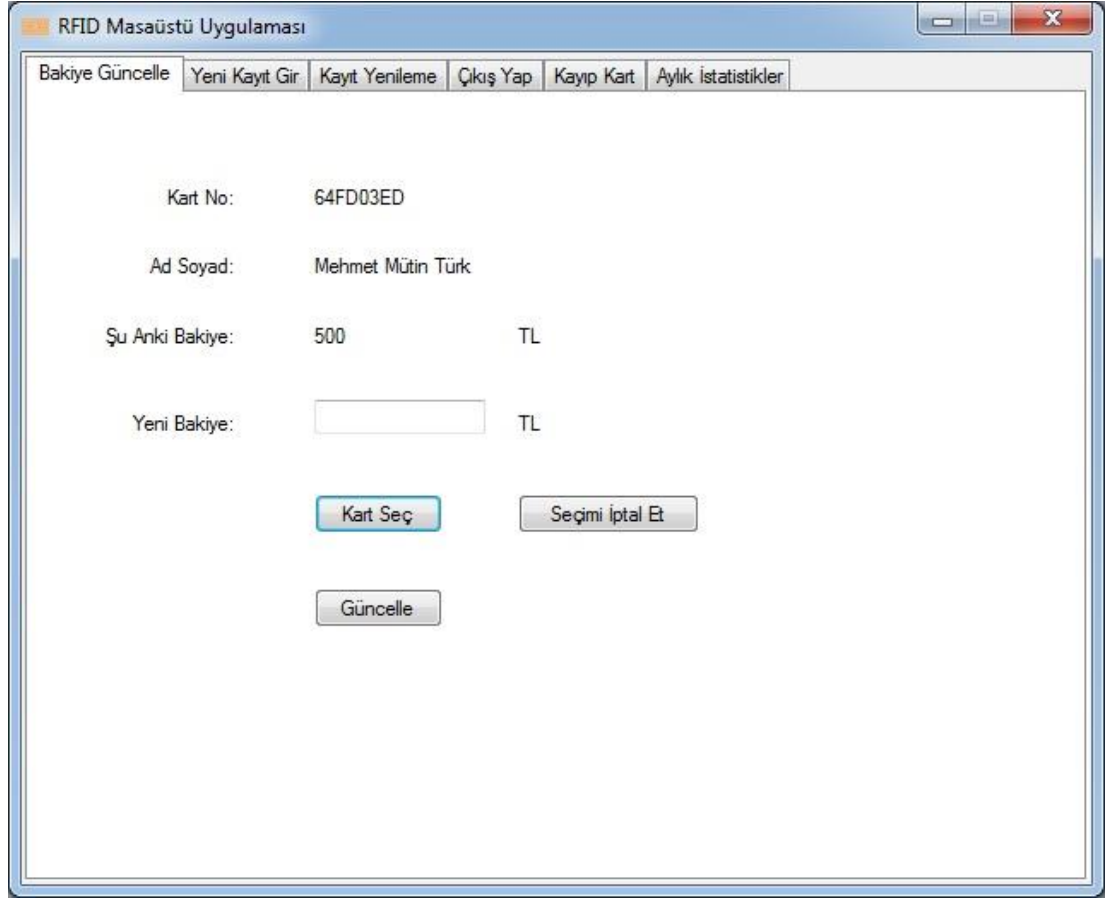
### 3.3.2. RFID Masaüstü Uygulaması

RFID Masaüstü Uygulaması işletmedeki kasada kullanılmaktadır. Kullanıcı grafik arayüzüne sahip bir uygulamadır. Kayıt ekleme, kayıt yenileme, bakiye güncelleme, kayıp kart ve müşteri çıkış işlemlerinden sorumludur. Bar ve havuzdan yapılan girişlerin sayısına ve bu girişlerin gün ve saat dağılımlarına da yine bu uygulama aracılığıyla erişilmektedir. Bu uygulamanın kaynak kodları EK 3'te verilmiştir.

The screenshot shows a Windows-style application window titled "RFID Masaüstü Uygulaması". The window has a menu bar with the following items: "Bakiye Güncelle", "Yeni Kayıt Gir", "Kayıt Yenileme", "Çıkış Yap", "Kayıp Kart", and "Aylık İstatistikler". The main content area is a form for updating a balance. It contains the following elements:

- "Kart No:" followed by a text input field.
- "Ad Soyad:" followed by a text input field.
- "Şu Anki Bakiye:" followed by a text input field and "TL" to its right.
- "Yeni Bakiye:" followed by a text input field and "TL" to its right.
- Below the input fields, there are three buttons: "Kart Seç", "Seçimi İptal Et", and "Güncelle".

Şekil 3.7. RFID Masaüstü Uygulaması bakiye güncelleme ekranı



Şekil 3.8. Kart seçili iken RFID Masaüstü Uygulaması bakiye güncelleme ekranı

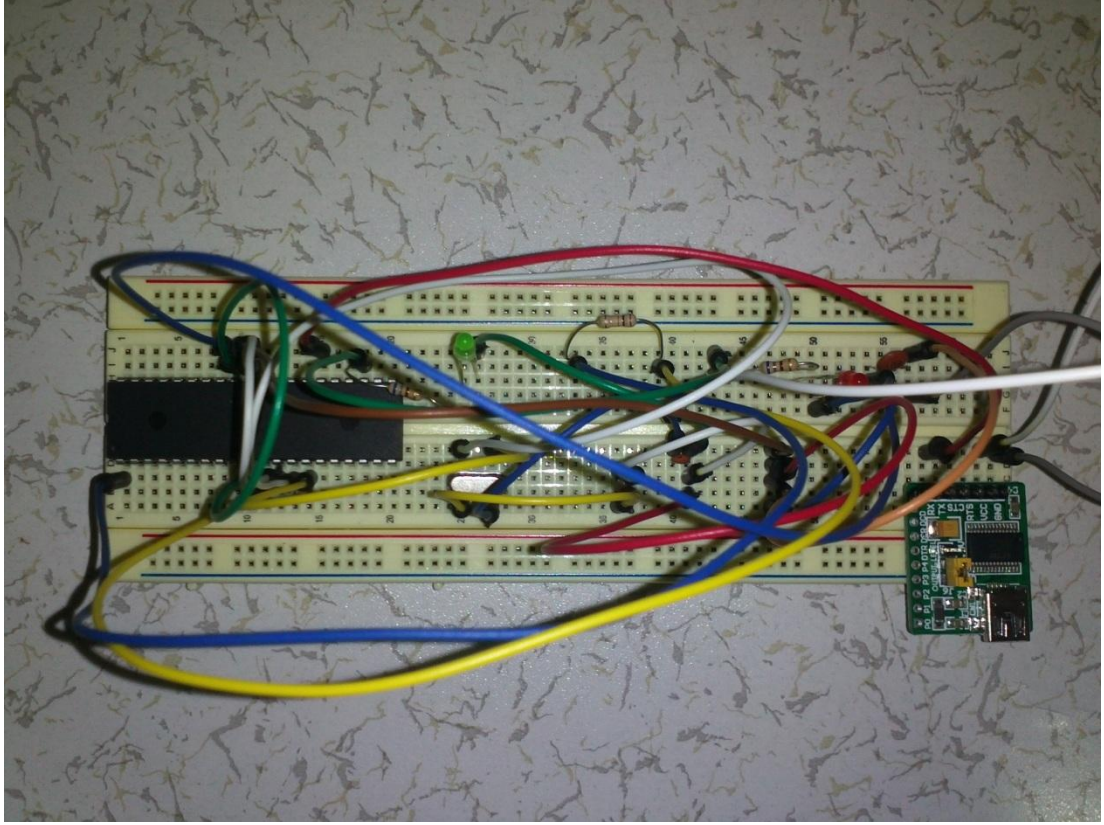
### 3.3.3. PIC C Uygulaması

Devrede Mikroçip firmasının üretmiş olduğu PIC 16F877A mikrodenetleyicisi kullanılmıştır. PIC ile bilgisayar arasındaki iletişim Mikroelektronika firmasının üretmiş olduğu USB UART kartı ile sağlanmaktadır. Bu kart USB seviyesindeki sinyalleri TTL (transistör-transistör mantığı) seviyesindeki sinyallere çevirmektedir. Kart çalışması için gerekli gücü USB portundan çekmektedir. Kartın üzerinde 5 voltluk bir gerilim çıkışı olduğundan, devrede aynı zamanda bir güç kaynağı olarak kullanılmıştır.

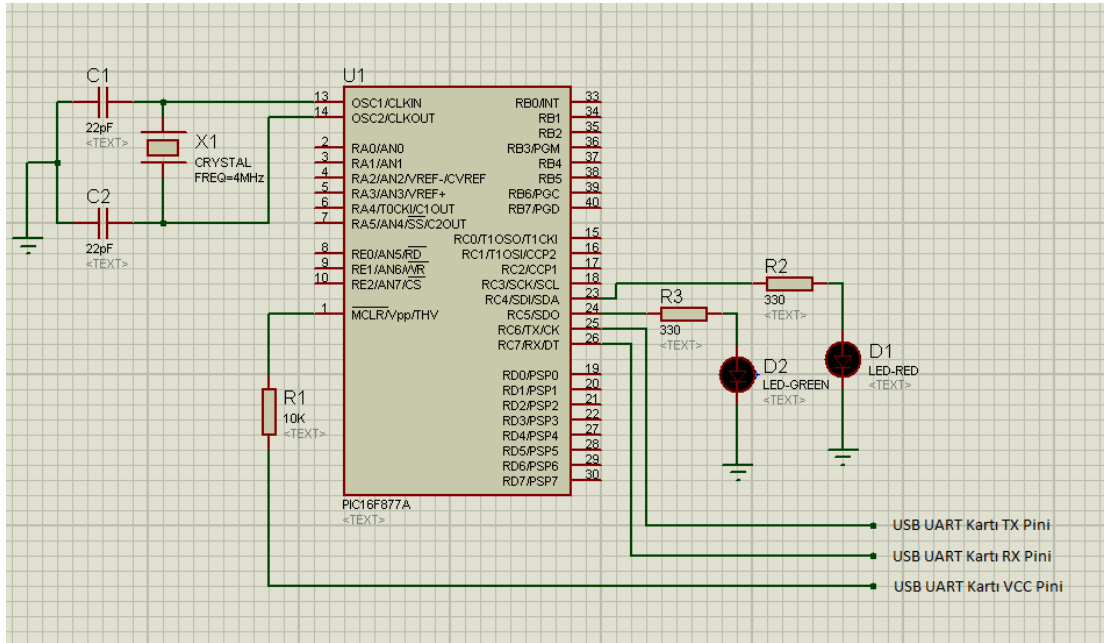


Şekil 3.9. Mikroelektronika USB UART kartı

Kontrol devresinde geçiş izninin olduğu durumlarda yeşil LED, geçiş izninin olmadığı durumlarda ise kırmızı LED yakılmaktadır. Geçiş izninin olduğu durumda bilgisayardan PIC'e 'y' karakteri, geçiş izninin olmadığı durumlarda ise 'n' karakteri gönderilmiştir.



Şekil 3.10. Kontrol devresi

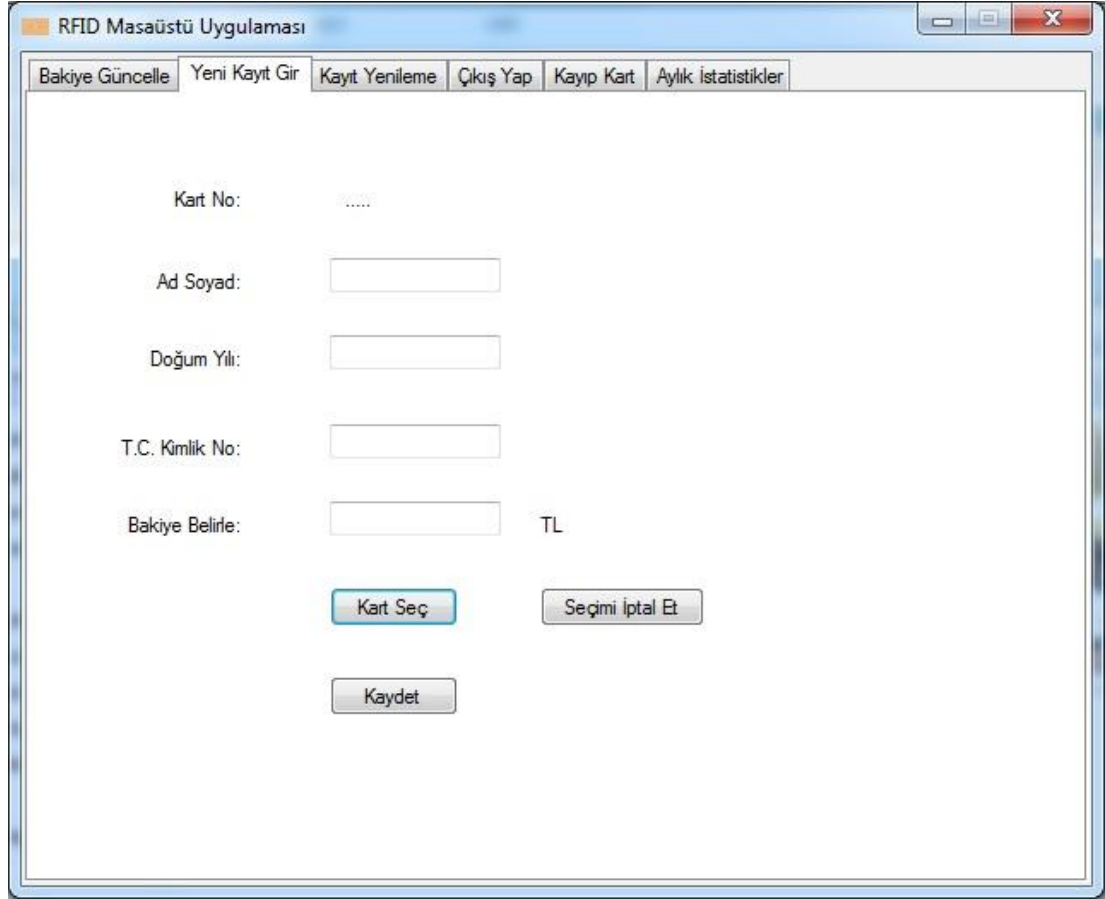


Şekil 3.11. Kontrol devresi devre şeması

PIC C Uygulaması'nın kaynak kodları EK 4'te verilmiştir.

#### 4. BULGULAR VE TARTIŞMA

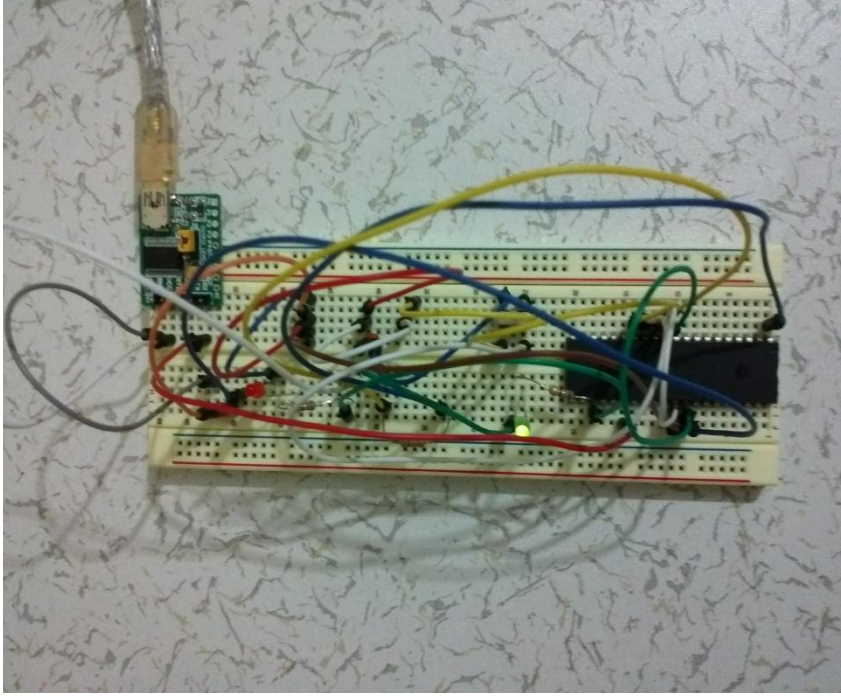
Bir müşteri işletmeye geldiği zaman, öncelikle kasada kaydı yapılmakta ve kendisine bir RFID etiketi verilmektedir. Müşteri bilgileri ve müşteriye verilen etiketin seri numarası RFID Masaüstü Uygulaması aracılığıyla veritabanına kaydedilmektedir.



Şekil 4.1. RFID Masaüstü Uygulaması müşteri kayıt ekranı

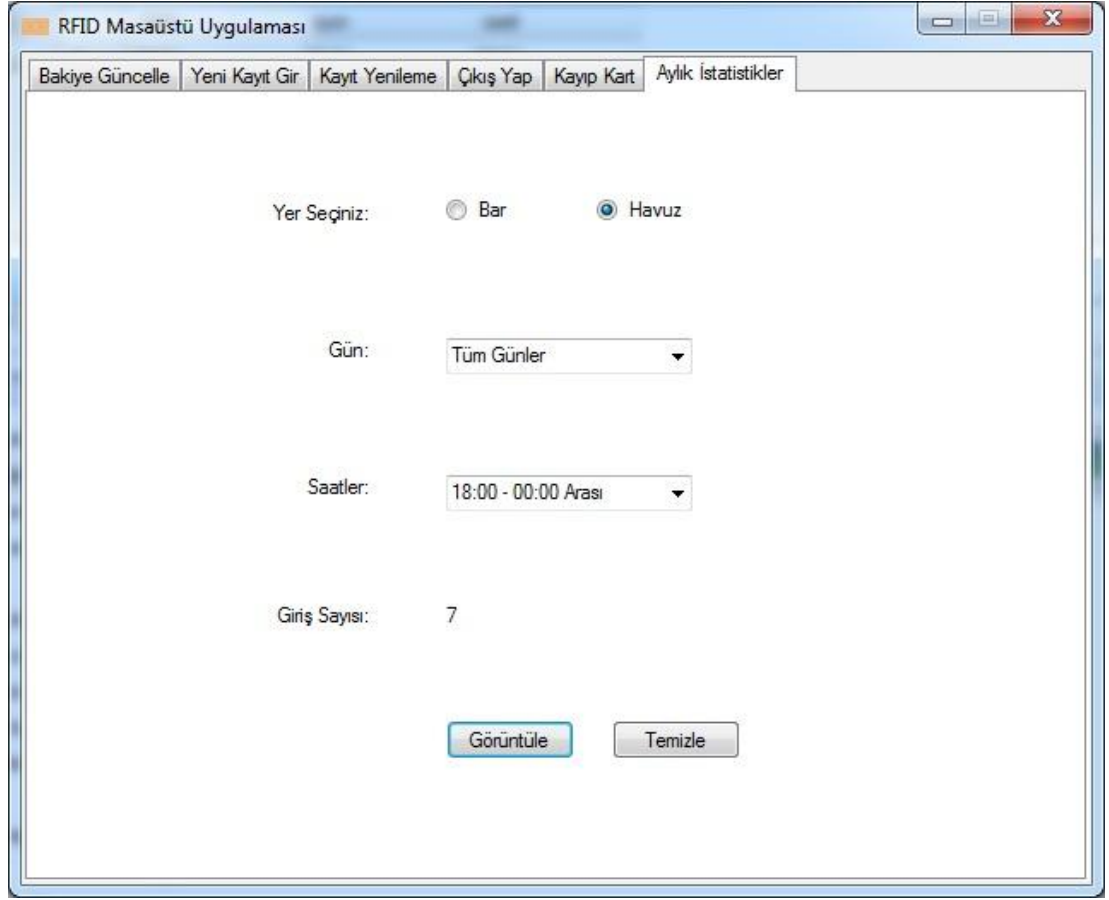
İşletmedeki havuza veya bara girmek isteyen müşteri kendisine verilen etiket aracılığıyla giriş yapabilmektedir. Bar girişinde, yaş kontrolü, veritabanındaki bilgiler kontrol edilerek otomatik olarak gerçekleşmektedir. Yaşı onsekizden büyük olan müşterinin, yeterli bakiyesi olması durumunda girişine izin verilmektedir. Giriş gerçekleştiği anda, müşterinin bakiyesi düşürülmektedir. Havuz girişinde ise yaş kontrolü yapılmamakta fakat barda olduğu gibi bir otomatik ödeme sistemi uygulanmaktadır. Bu işlemler RFID Konsol Uygulaması tarafından kontrol edilmektedir.

Bar ve havuz girişleri, kontrol devresi aracılığıyla simüle edilmiştir. Devrede bulunan LED'ler bölüm girişlerindeki kapıları temsil etmektedir. Kırmızı LED'in yakılması kapının açılmadığı durumu, yeşil LED ise kapının açıldığı durumu göstermektedir.



Şekil 4.2. Başarılı geçiş durumunda kontrol devresi

Başarılı geçiş durumlarında, geçişin olduğu tarih, gün ve saat veritabanına kaydedilmektedir. Kaydedilen bilgilere RFID Masaüstü Uygulamasının aylık istatistikler bölümünden ulaşmak mümkündür. İşletmenin, bu bilgiler çerçevesinde daha isabetli kararlar verebilmesine olanak sağlanmıştır.



Şekil 4.3. RFID Masaüstü Uygulaması aylık istatistikler ekranı



## **5. SONUÇLAR VE ÖNERİLER**

Bu çalışmadaki otomatik kontrol ve ödeme sisteminin, ticari işletmelerde zaman ve hizmet maliyetlerini önemli ölçüde azaltacağı düşünülmektedir. İşletmede, bölüm girişlerinde kayıt alınması ise işletmenin daha isabetli yatırım kararları vermesine olanak sağlamaktadır.

Oluşturulan RFID sisteminde, okuyucular ile bilgisayar arasındaki bağlantı kablolar aracılığıyla sağlanmıştır. Kurulum aşamasında olan bir işletmede böyle bir RFID sistemi kurulması düşünülebilir. Hizmet aşamasında olan bir işletmede ise tamamen kablosuz bir RFID sistemi kurulmasının daha isabetli olacağı düşünülmektedir.

Geliştirilen sistem ticari işletmelerdeki tüm bölümlere kolaylıkla uyarlanabilir. Örneğin, otel odalarına girişlerin etiketler ile yapılması, işletmedeki diğer bölümlere farklı bir ödeme sistemi uygulanması gibi işlemler de kolaylıkla gerçekleştirilebilir.

Müşteri işlemlerinin yanısıra personel işlemleri de kontrol edilebilir. Personellerin işletmede yalnızca belirlenen bölümlere giriş izinlerinin sağlanması, personellerin işletmeye giriş – işletmeden çıkış saatlerinin kaydı gibi işlemler de bu çalışma baz alınarak gerçekleştirilebilir.

Bu çalışmada kullanılan otomatik ödeme sistemi; toplu ulaşım hizmetlerine, müzelere, eğlence merkezlerine ve benzeri birçok sisteme kolaylıkla uyarlanabilir.

## **KAYNAKLAR**

Albahari, J., Albahari, B., “C# 5.0 in a Nutshell The Definitive Reference”, O’Reilly Media, California, 1064 s., (2012).

Çiçek, S., “CCS C İle PIC Programlama”, Altaş Yayıncılık, İstanbul, 600 s., (2009).

Glover, B., Bhatt, H., “RFID Essentials”, O’Reilly Media, California, 288 s., (2006).

Lahiri, S., “RFID Sourcebook”, IBM Press, Indianapolis, 304 s., (2005).

Müller, D. (Translated by), “RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication, 3rd Edition”, Wiley, New York City, 478 s., (2010).

[Finkenzeller, K., “RFID Handbuch: Grundlagen und praktische Anwendungen induktiver Funkanlagen, Transponder und kontaktloser Chipkarten”, Carl Hansen Verlag, München, (2008).]

Üstündağ, A., “RFID ve Tedarik Zinciri”, İTÜ RFID Laboratuvarı,

<http://www.rfid.itu.edu.tr/arastirma/RFID-rfidfuari08sunum.pdf>  
(29.08.2008)

## EK 1. mifare.cs Kaynak Kodları

```
using System;
using System.IO.Ports;
using sm_mifare_lib;

namespace sm_mifare_lib
{
    public enum ASource
    {
        KeyTypeA = 0xAA,
        KeyTypeB = 0xBB,
        KeyMifareDefault = 0xFF
    }

    public class mifare
    {
        public static readonly byte[] E2promKeyA = new byte[16] { 0x10, 0x11,
0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19, 0x1A, 0x1B, 0x1C, 0x1D, 0x1E,
0x1F };
        public static readonly byte[] E2promKeyB = new byte[16] { 0x20, 0x21,
0x22, 0x23, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x2A, 0x2B, 0x2C, 0x2D, 0x2E,
0x2F };

        public SerialPort port = new SerialPort();

        private bool get_response(out byte[] MyBuffer, out byte Count)
        {
            MyBuffer = new byte[24];
            int i;
            Count = 0;

            //Get first 4 bytes from the device, this includes command and
length information of data.
            for (i = 0; i < 4; i++)
            {
                try
                {
                    MyBuffer[i] = (byte)port.ReadByte();
                }
                catch (Exception)
                {
                    return false;
                }
            }

            if (MyBuffer[2] > 20) //Mybuffer[2] represents length of
data+checksum. This can not be greater than 20.
                return false;

            //Get Remaining data.
            for (i = 0; i < MyBuffer[2]; i++)
            {
                try
                {
                    MyBuffer[i + 4] = (byte)port.ReadByte();
                }
                catch (Exception)
            }
        }
    }
}
```

```
        {
            return false;
        }
    }

    Count = MyBuffer[2];
    Count += 4;
    return true;
}

public void get_data(ref byte[] Response, ref byte Count) //This
will filter only data portion of the whole incoming data(Remove
Header,Command,Length,Checksum).
{
    int i;
    byte y;
    y = 0;

    if (Count >= 4)//If Count<4 then this is not a valid packet.
    {
        Count--; //We won't use final byte that is Checksum.

        for (i = 4; i < Count; i++)
        {
            Response[y] = Response[i];
            y++;
        }
    }

    Count = y;
}

public void send_command(byte Command, byte[] DataBuffer, byte
DataCount)
{
    byte[] TxFrame = new byte[24];
    int i;
    byte checksum;

    DataCount++; //Length information will include checksum byte.
    checksum = 0;
    TxFrame[0] = 0xFF;
    TxFrame[1] = 0x00;
    TxFrame[2] = DataCount;
    TxFrame[3] = Command;

    checksum = TxFrame[1];
    checksum += TxFrame[2];
    checksum += TxFrame[3];

    for (i = 0; i < DataCount; i++)
    {
        TxFrame[i + 4] = DataBuffer[i];
        checksum += DataBuffer[i];
    }

    TxFrame[DataCount + 3] = checksum;
    port.Write(TxFrame, 0, DataCount + 4);
}
```

```
public void ParseIncoming(out byte Response_Type, out byte TagType,
out byte[] TagSerial, out string Firmware)
{
    byte[] Response = new byte[24];
    byte Count = 0;
    byte i = 0;

    TagSerial = new byte[4];
    Firmware = "";
    Response_Type = 0;
    TagType = 0;

    if (get_response(out Response, out Count))
    {
        if (Response[3] == 0x81) //Is this Firmware Version?
        {
            get_data(ref Response, ref Count); //This will filter only
data portion of the whole incoming data(Remove
Header,Command,Length,Checksum).

            //Create Firmware String.
            for (i = 0; i < Count; i++)
                Firmware += Convert.ToChar(Response[i]);

            Response_Type = 1;
        }
        else if (Response[3] == 0x82) // Is this Tag type and Serial?
        {
            get_data(ref Response, ref Count); //This will filter only
data portion of the whole incoming data(Remove
Header,Command,Length,Checksum).

            TagType = Response[0]; //Tag Type. i.e
0x01 = Mifare 1K, 0x02 = Mifare 4K.
            TagSerial[0] = Response[4]; //Serial number comes
LSB first.

            TagSerial[1] = Response[3];
            TagSerial[2] = Response[2];
            TagSerial[3] = Response[1];

            Response_Type = 2;
        }
    }
}

public bool OpenPort(string PortName, int BaudRate)
{
    //Open port if not opened previously.
    if (port.IsOpen)
        port.Close();

    if (!port.IsOpen)
    {
        port.PortName = PortName;
        port.BaudRate = BaudRate;
        port.ReadTimeout = 1000;
        port.WriteTimeout = 500;
        port.DtrEnable = true;
        try
        {
```

```
        port.Open();
    }
    catch (Exception) //catch (Exception ee).
    {
        return false;
    }
    port.ReadExisting(); //Just to clear Buffer.
}

    return true;
}

public void ClosePort()
{
    //Open port if not opened previously.
    if (port.IsOpen)
    {
        port.Close();
    }
}

public bool CMD_ResetDevice(out string Firmware)
{
    byte[] Response = new byte[24];
    byte[] DataBuffer = new byte[20];
    int i;
    byte Count = 0;
    Firmware = "";

    port.ReadExisting(); //Just to clear Buffer.
    send_command(0x80, DataBuffer, 0);

    //We wait reset_response after a reset.
    //That is the firmware version of the module.
    if (get_response(out Response, out Count))
    {
        get_data(ref Response, ref Count); //This will filter only
data portion of the whole incoming data(Remove
Header,Command,Length,Checksum).

        //Create Firmware String.
        for (i = 0; i < Count; i++)
            Firmware += Convert.ToChar(Response[i]);

        return true;
    }
    else
        return false;
}

public bool CMD_SelectTag(out byte TagType, out byte[] TagSerial, out
byte ReturnCode)
{
    byte[] Response = new byte[24];
    byte[] DataBuffer = new byte[20];
    TagSerial = new byte[4];
    byte Count = 0;
    TagType = 0;

    ReturnCode = 0; // 0 Communication Error.
```

```
port.ReadExisting(); //Just to clear Buffer.
send_command(0x83, DataBuffer, 0); // 0x83 = SELECT_TAG.

//Wait response of the operation.
if (get_response(out Response, out Count))
{
    get_data(ref Response, ref Count); //This will filter only
data portion of the whole incoming data(Remove
Header,Command,Length,Checksum).

    if (Count > 1)
    {
        TagType = Response[0]; //Tag Type. i.e
0x01 = Mifare 1K, 0x02 = Mifare 4K.
        TagSerial[0] = Response[4]; //Serial number comes
LSB first.
        TagSerial[1] = Response[3];
        TagSerial[2] = Response[2];
        TagSerial[3] = Response[1];
        return true; //1-> Success.
    }
    else
    {
        ReturnCode = Response[0];
        return false;
    }
}
else
return false;
}

public bool CMD_SeekForTag(out byte ReturnCode)
{
    byte[] Response = new byte[24];
    byte[] DataBuffer = new byte[20];
    byte Count = 0;
    ReturnCode = 0;

    port.ReadExisting(); //Just to clear Buffer.
    send_command(0x82, DataBuffer, 0); // 0x83 = SELECT_TAG.

    //Wait response of the operation.
    if (get_response(out Response, out Count))
    {
        get_data(ref Response, ref Count); //This will
filter only data portion of the whole incoming data(Remove
Header,Command,Length,Checksum).

        if (Response[0] == 0x4C)
            return true;
        else
        {
            ReturnCode = Response[0];
            return false;
        }
    }
    else
        return false;
}
```

```
public bool CMD_Authenticate(byte AuthSource, byte[] Key, byte
BlockNo, out byte ReturnCode)
{
    byte[] Response = new byte[24];
    byte[] DataBuffer = new byte[20];
    byte i = 0;
    byte Count = 0;
    ReturnCode = 0;

    DataBuffer[0] = BlockNo;
    DataBuffer[1] = AuthSource;

    if ((AuthSource == (byte)ASource.KeyTypeA) || (AuthSource ==
(byte)ASource.KeyTypeB))
    {
        for (i = 0; i < 6; i++)
            DataBuffer[2 + i] = Key[i];
        send_command(0x85, DataBuffer, 8);
    }
    else
        send_command(0x85, DataBuffer, 2);

    if (get_response(out Response, out Count))
    {
        get_data(ref Response, ref Count); //This will
filter only data portion of the whole incoming data(Remove
Header,Command,Length,Checksum).

        if (Response[0] == 0x4C)
            return true;
        else
        {
            ReturnCode = Response[0];
            return false;
        }
    }
    else
        return false;
}

public bool CMD_Halt(out byte ReturnCode)
{
    byte[] Response = new byte[24];
    byte[] DataBuffer = new byte[20];
    byte Count = 0;
    ReturnCode = 0;

    port.ReadExisting(); //Just to clear Buffer.
    send_command(0x93, DataBuffer, 0); // 0x83 = SELECT_TAG.

    //Wait response of the operation.
    if (get_response(out Response, out Count))
    {
        get_data(ref Response, ref Count); //This will
filter only data portion of the whole incoming data(Remove
Header,Command,Length,Checksum).

        if (Response[0] == 0x4C)
            return true;
    }
}
```



```
        else
        {
            ReturnCode = Response[0];
            return false;
        }
    }
    else
        return false;
}

public bool CMD_ReadBlock(byte BlockNo, out byte[] BlockData, out byte
ReturnCode)
{
    byte[] Response = new byte[24];
    byte[] DataBuffer = new byte[20];
    BlockData = new byte[16];
    byte Count = 0;
    byte i = 0;

    ReturnCode = 0;        //0 Communication Error.

    DataBuffer[0] = BlockNo;

    port.ReadExisting();        //Just to clear Buffer.
    send_command(0x86, DataBuffer, 1); // 0x86 = Read Block.

    //Wait response of the operation.
    if (get_response(out Response, out Count))
    {
        get_data(ref Response, ref Count);        //This will
filter only data portion of the whole incoming data(Remove
Header,Command,Length,Checksum).
        if (Count <= 1)
            //Means Error.
            {
                ReturnCode = Response[0];
                return false;
            }
        else
            {
                for (i = 0; i < 16; i++)
                {
                    BlockData[i] = Response[i + 1];
                    ReturnCode = Response[0];
                }
                return true;
            }
    }
    else
        return false;
}

private bool check_if_sector_trailer(byte BlockNo)
{
    int remain = 0;
    int temp = 0;

    if (BlockNo <= 128)
    {
```

```
        temp = BlockNo;
        temp++;
        remain = temp % 4;

        if (remain == 0)
        {
            return false;
        }
    }
    else
    {
        temp = BlockNo;
        temp++;
        remain = temp % 16;

        if (remain == 0)
        {
            return false;
        }
    }
    return true;
}

public bool CMD_WriteBlock(byte BlockNo, byte[] BlockData, out byte
ReturnCode)
{
    byte[] Response = new byte[24];
    byte[] DataBuffer = new byte[20];
    byte Count = 0;
    byte i = 0;

    ReturnCode = 0; //0 Communication Error.

    if (!check_if_sector_trailer(BlockNo))
    {
        ReturnCode = 1; //This is Sector Trailer block. Write
to Sector Trailer block is not allowed with this function.
        return false;
    }

    DataBuffer[0] = BlockNo;

    for (i = 0; i < 16; i++)
        DataBuffer[i + 1] = BlockData[i];

    port.ReadExisting(); //Just to clear Buffer.
    send_command(0x89, DataBuffer, 17); // 0x89 = Write Block.

    //Wait response of the operation.
    if (get_response(out Response, out Count))
    {
        get_data(ref Response, ref Count); //This will
filter only data portion of the whole incoming data(Remove
Header,Command,Length,Checksum).

        if (Count <= 1)
//Means Error.
        {
            ReturnCode = Response[0];
            return false;
        }
    }
}
```

```
    }
    else
    {
        //Here we have 16 bytes of written data. But we won't
process it.
        //No need to send back what data is written on tag. Module
already written and confirmed written data with read operation.
        return true;
    }
}
else
return false;
}

public bool CMD_ReadValue(byte BlockNo, out long Value, out byte
ReturnCode)
{
    byte[] Response = new byte[24];
    byte[] DataBuffer = new byte[20];
    byte Count = 0;

    Value = 0;
    ReturnCode = 0;           //0 Communication Error.

    DataBuffer[0] = BlockNo;

    port.ReadExisting();           //Just to clear Buffer.
    send_command(0x87, DataBuffer, 1); // 0x86 = Read Block.

    //Wait response of the operation.
    if (get_response(out Response, out Count))
    {
        get_data(ref Response, ref Count);           //This will
filter only data portion of the whole incoming data(Remove
Header,Command,Length,Checksum).

        if (Count < 5)
//Means Error.
        {
            ReturnCode = Response[0];
            return false;
        }
        else
        {
            Value = BitConverter.ToInt32(Response, 1);
            return true;
        }
    }
    else
return false;
}

public bool CMD_WriteValue(byte BlockNo, long Value, out byte
ReturnCode)
{
    byte[] Response = new byte[24];
    byte[] DataBuffer = new byte[20];
    byte[] ValueArray = new byte[8];
    byte Count = 0;
    byte i = 0;
```

```
        ReturnCode = 0;                //0 Communication Error.

        if (!check_if_sector_trailer(BlockNo))
        {
            ReturnCode = 1;            //This is Sector Trailer block.
            Sector Trailer block can not be Value Block.
            return false;
        }

        DataBuffer[0] = BlockNo;
        ValueArray = BitConverter.GetBytes(Value);

        for (i = 0; i < 4; i++)
            DataBuffer[i + 1] = ValueArray[i];

        port.ReadExisting();            //Just to clear Buffer.
        send_command(0x8A, DataBuffer, 5); // 0x8A = Write Value.

        //Wait response of the operation.
        if (get_response(out Response, out Count))
        {
            get_data(ref Response, ref Count);            //This will
            filter only data portion of the whole incoming data(Remove
            Header,Command,Length,Checksum).

            if (Count < 5)
                //Means Error.
                {
                    ReturnCode = Response[0];
                    return false;
                }
            else
                {
                    //No need to send back the read value, as module already
                    compared it with written value.
                    return true;
                }
        }
        else
            return false;
    }

    public bool CMD_IncrementValue(byte BlockNo, long IncValue, out long
    NewValue, out byte ReturnCode)
    {
        byte[] Response = new byte[24];
        byte[] DataBuffer = new byte[20];
        byte[] ValueArray = new byte[8];
        byte Count = 0;
        byte i = 0;

        NewValue = 0;
        ReturnCode = 0;                //0 Communication Error.

        DataBuffer[0] = BlockNo;
        ValueArray = BitConverter.GetBytes(IncValue);

        for (i = 0; i < 4; i++)
            DataBuffer[i + 1] = ValueArray[i];
    }
}
```

```
port.ReadExisting(); //Just to clear Buffer.
send_command(0x8D, DataBuffer, 5); // 0x8D = Increment Value.

//Wait response of the operation.
if (get_response(out Response, out Count))
{
    get_data(ref Response, ref Count); //This will
filter only data portion of the whole incoming data(Remove
Header,Command,Length,Checksum).

    if (Count < 5) //Means Error.
    {
        ReturnCode = Response[0];
        return false;
    }
    else
    {
        NewValue = BitConverter.ToInt32(Response, 1);
        return true;
    }
}
else
return false;
}

public bool CMD_DecrementValue(byte BlockNo, long DecValue, out long
NewValue, out byte ReturnCode)
{
    byte[] Response = new byte[24];
    byte[] DataBuffer = new byte[20];
    byte[] ValueArray = new byte[8];
    byte Count = 0;
    byte i = 0;

    NewValue = 0;
    ReturnCode = 0; //0 Communication Error.

    DataBuffer[0] = BlockNo;
    ValueArray = BitConverter.GetBytes(DecValue);

    for (i = 0; i < 4; i++)
        DataBuffer[i + 1] = ValueArray[i];

    port.ReadExisting(); //Just to clear Buffer.
    send_command(0x8E, DataBuffer, 5); // 0x8E = Decrement Value.

    //Wait response of the operation.
    if (get_response(out Response, out Count))
    {
        get_data(ref Response, ref Count); //This will
filter only data portion of the whole incoming data(Remove
Header,Command,Length,Checksum).

        if (Count < 5) //Means Error.
        {
            ReturnCode = Response[0];
            return false;
        }
        else
        {
            NewValue = BitConverter.ToInt32(Response, 1);
            return true;
        }
    }
    else
    {
        ReturnCode = 0;
        return false;
    }
}
```

```
        {
            NewValue = BitConverter.ToInt32(Response, 1);
            return true;
        }
    }
    else
        return false;
}

public bool CMD_SwitchRF(bool OnOffState, out byte ReturnCode)
{
    byte[] Response = new byte[24];
    byte[] DataBuffer = new byte[20];
    byte Count = 0;

    ReturnCode = 0; //0 Communication Error.

    if (OnOffState)
        DataBuffer[0] = 0xFF;
    else
        DataBuffer[0] = 0;

    port.ReadExisting(); //Just to clear Buffer.
    send_command(0x90, DataBuffer, 1); // 0x90 = Switch RF ON/OFF
    Command.

    //Wait response of the operation.
    if (get_response(out Response, out Count))
    {
        get_data(ref Response, ref Count); //This will
        filter only data portion of the whole incoming data(Remove
        Header,Command,Length,Checksum).

        if (Count == 1) //Success.
        {
            if ((Response[0] == 0) || (Response[0] == 1))
                return true;
            else
            {
                ReturnCode = Response[0];
                return false;
            }
        }
        else
        {
            ReturnCode = Response[0];
            return false;
        }
    }
    else
        return false;
}
}
}
```

## EK 2. RFID Konsol Uygulaması Kaynak Kodları

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using sm_mifare_lib;
using System.Data;
using System.Data.SqlServerCe;
using System.Threading;
using System.IO.Ports;

namespace RFID_Konsol_Uygulaması
{
    class Program
    {
        static mifare barreader = new mifare();
        static mifare havuzreader = new mifare();
        static string barport = "COM4";
        static string havuzport = "COM5";
        static string picport = "COM6";
        static SerialPort port = new SerialPort(picport, 9600, Parity.None, 8,
StopBits.One);
        static string dbpath = @"Data Source = |DataDirectory|\Database.sdf";

        static void Main(string[] args)
        {
            Thread barthread = new Thread(barreader_method);
            Thread havuzthread = new Thread(havuzreader_method);
            Thread dbresetthread = new Thread(dbreset_method);
            barthread.Start();
            havuzthread.Start();
            dbresetthread.Start();
        }

        static void barreader_method()
        {
            barreader.OpenPort(barport, 19200);
            byte TagType;
            byte[] TagSerial = new byte[4];
            byte ReturnCode = 0;
            string bartagno;
            List<Tuple<DateTime, String>> previoustags = new
List<Tuple<DateTime, String>>();
            DateTime lastreaddt = DateTime.MinValue;

            while (true)
            {
                bartagno = "";
                while (!barreader.CMD_SelectTag(out TagType, out TagSerial,
out ReturnCode))
                {
                }
                for (int i = 0; i < 4; i++)
                {
                    bartagno += TagSerial[i].ToString("X2");
                }
                if (DateTime.Now - lastreaddt > TimeSpan.FromMinutes(1))
```

```
        {
            previoustags.Clear();
            barislem(bartagno);
            previoustags.Add(Tuple.Create(DateTime.Now, bartagno));
            lastreaddt = DateTime.Now;
        }
        else
        {
            if (!previoustags.Exists(a => a.Item2.Equals(bartagno)))
            {
                barislem(bartagno);
                previoustags.Add(Tuple.Create(DateTime.Now,
                bartagno));
                lastreaddt = DateTime.Now;
            }
        }
        previoustags.RemoveAll(a => a.Item1.CompareTo(DateTime.Now -
        TimeSpan.FromMinutes(1)) < 0);
        Thread.Sleep(750);
    }
}

static void havuzreader_method()
{
    havuzreader.OpenPort(havuzport, 19200);
    byte TagType;
    byte[] TagSerial = new byte[4];
    byte ReturnCode = 0;
    string havuztagno;
    List<Tuple<DateTime, String>> previoustags = new
    List<Tuple<DateTime, String>>();
    DateTime lastreaddt = DateTime.MinValue;

    while (true)
    {
        havuztagno = "";
        while (!havuzreader.CMD_SelectTag(out TagType, out TagSerial,
        out ReturnCode))
        {
        }
        for (int i = 0; i < 4; i++)
        {
            havuztagno += TagSerial[i].ToString("X2");
        }
        if (DateTime.Now - lastreaddt > TimeSpan.FromMinutes(1))
        {
            previoustags.Clear();
            havuzislem(havuztagno);
            previoustags.Add(Tuple.Create(DateTime.Now, havuztagno));
            lastreaddt = DateTime.Now;
        }
        else
        {
            if (!previoustags.Exists(a => a.Item2.Equals(havuztagno)))
            {
                havuzislem(havuztagno);
                previoustags.Add(Tuple.Create(DateTime.Now,
                havuztagno));
                lastreaddt = DateTime.Now;
            }
        }
    }
}
```



```
    }
    previoustags.RemoveAll(a => a.Item1.CompareTo(DateTime.Now -
TimeSpan.FromMinutes(1)) < 0);
    Thread.Sleep(750);
}
}

static void dbreset_method()
{
    while (true)
    {
        Thread.Sleep(86400000);
        dbreset_islem();
    }
}

static void barislem(string bartagno)
{
    SqlConnection conn = new SqlConnection(dbpath);
    try
    {
        conn.Open();
        int dogumyili;
        string tagno = bartagno;
        int bakiye;
        SqlCommand cmd1 = new SqlCommand("select dogumyili from
musteritable where tagno = @tagno", conn);
        cmd1.Parameters.Add("@tagno", SqlDbType.NVarChar).Value =
tagno;

        SqlDataReader rdr1 = cmd1.ExecuteReader();
        rdr1.Read();
        dogumyili = (int)rdr1["dogumyili"];

        int yas = DateTime.Now.Year - dogumyili;
        if (yas >= 18)
        {
            SqlCommand cmd2 = new SqlCommand("select bakiye from
musteritable where tagno = @tagno", conn);
            cmd2.Parameters.Add("@tagno", SqlDbType.NVarChar).Value =
tagno;

            SqlDataReader rdr2 = cmd2.ExecuteReader();
            rdr2.Read();
            bakiye = (int)rdr2["bakiye"];

            if (bakiye >= 10)
            {
                gecisserbest();
                tarihkaydibar();
                int newbakiye = bakiye - 10;

                SqlCommand cmd3 = new SqlCommand("update
musteritable set bakiye = @newbakiye where tagno = @tagno");
                cmd3.Connection = conn;
                cmd3.Parameters.Add("@newbakiye", SqlDbType.Int).Value =
newbakiye;
                cmd3.Parameters.Add("@tagno",
SqlDbType.NVarChar).Value = tagno;
                cmd3.ExecuteNonQuery();
            }
            else

```

```
        {
            gecisyok();
        }
    }
    else
    {
        gecisyok();
    }
}
catch
{
}
finally
{
    conn.Close();
}
}

static void havuzislem(string havuztagno)
{
    SqlConnection conn = new SqlConnection(dbpath);
    try
    {
        conn.Open();
        int bakiye;
        string tagno = havuztagno;
        SqlCommand cmd1 = new SqlCommand("select bakiye from
musteritable where tagno = @tagno", conn);
        cmd1.Parameters.Add("@tagno", SqlDbType.NVarChar).Value =
tagno;

        SqlDataReader rdr1 = cmd1.ExecuteReader();
        rdr1.Read();
        bakiye = (int)rdr1["bakiye"];

        if (bakiye >= 10)
        {
            gecisserbest();
            tarihkaydihavuz();
            int newbakiye = bakiye - 10;

            SqlCommand cmd2 = new SqlCommand("update musteritable
set bakiye = @newbakiye where tagno = @tagno");
            cmd2.Connection = conn;
            cmd2.Parameters.Add("@newbakiye", SqlDbType.Int).Value =
newbakiye;
            cmd2.Parameters.Add("@tagno", SqlDbType.NVarChar).Value =
tagno;

            cmd2.ExecuteNonQuery();
        }
        else
        {
            gecisyok();
        }
    }
    catch
    {
    }
    finally
    {
        conn.Close();
    }
}
```

```
    }  
  }  
  
  static void dbreset_islem()  
  {  
    SqlConnection conn = new SqlConnection(dbpath);  
    try  
    {  
      conn.Open();  
      SqlCommand cmd = new SqlCommand("delete from bartable  
where date < getdate() - 30");  
      cmd.Connection = conn;  
      cmd.ExecuteNonQuery();  
    }  
    finally  
    {  
      conn.Close();  
    }  
  }  
  
  static void gecisserbest()  
  {  
    if (!port.IsOpen)  
    {  
      port.Open();  
    }  
  
    port.Write("y");  
  }  
  
  static void gecisyok()  
  {  
    if (!port.IsOpen)  
    {  
      port.Open();  
    }  
  
    port.Write("n");  
  }  
  
  static void tarihkaydibar()  
  {  
    DateTime dt = DateTime.Now;  
    string gun = dt.ToString("dddd", new  
System.Globalization.CultureInfo("tr-TR"));  
    string saat = saatcevir(dt);  
    SqlConnection conn = new SqlConnection(dbpath);  
  
    try  
    {  
      conn.Open();  
      SqlCommand cmd = new SqlCommand("insert into bartable  
(date, gun, saat) values (@date, @gun, @saat)", conn);  
      cmd.Parameters.AddWithValue("@date", dt);  
      cmd.Parameters.AddWithValue("@gun", gun);  
      cmd.Parameters.AddWithValue("@saat", saat);  
      cmd.ExecuteNonQuery();  
    }  
    finally  
    {  

```

```
        conn.Close();
    }
}

static void tarih kaydihavuz()
{
    DateTime dt = DateTime.Now;
    string gun = dt.ToString("dddd", new
System.Globalization.CultureInfo("tr-TR"));
    string saat = saatcevir(dt);
    SqlConnection conn = new SqlConnection(dbpath);

    try
    {
        conn.Open();
        SqlCommand cmd = new SqlCommand("insert into havuztable
(date, gun, saat) values (@date, @gun, @saat)", conn);
        cmd.Parameters.AddWithValue("@date", dt);
        cmd.Parameters.AddWithValue("@gun", gun);
        cmd.Parameters.AddWithValue("@saat", saat);
        cmd.ExecuteNonQuery();
    }
    finally
    {
        conn.Close();
    }
}

static string saatcevir(DateTime dt)
{
    string sabah = "06:00 - 12:00 Arası";
    string oglen = "12:00 - 18:00 Arası";
    string akşam = "18:00 - 00:00 Arası";
    string gece = "00:00 - 06:00 Arası";
    string saat = dt.Hour.ToString();
    int saatint = Int32.Parse(saat);
    if (saatint >= 6 && saatint < 12)
    {
        saat = sabah;
    }
    else if (saatint >= 12 && saatint < 18)
    {
        saat = oglen;
    }
    else if (saatint >= 18)
    {
        saat = akşam;
    }
    else
    {
        saat = gece;
    }
    return saat;
}
}
```

### EK 3. RFID Masaüstü Uygulaması Kaynak Kodları

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlServerCe;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using sm_mifare_lib;

namespace RFID_Uygulaması
{
    public partial class Form1 : Form
    {
        mifare kasardr = new mifare();
        string kasardrport = "COM3";
        string dbpath = @"Data Source = |DataDirectory|\Database.sdf";

        public Form1()
        {
            InitializeComponent();
        }

        public string tagnoal()
        {
            kasardr.OpenPort(kasardrport, 19200);
            byte TagType;
            byte[] TagSerial = new byte[4];
            byte ReturnCode = 0;

            string tagno = "";
            while (!kasardr.CMD_SelectTag(out TagType, out TagSerial, out
ReturnCode)) //wait until a tag is present
            {
            }
            for (int i = 0; i < 4; i++)
            {
                tagno += TagSerial[i].ToString("X2");
            }
            return tagno;
        }

        private void kartsecbtn1_Click(object sender, EventArgs e)
        {
            try
            {
                string tagno = tagnoal();
                tagnolbl1.Text = tagno;
                SqlCeConnection conn = new SqlCeConnection(dbpath);
                try
                {
                    conn.Open();
                    SqlCeCommand cmd = new SqlCeCommand("select adsoyad,
bakiye from musteritable where tagno = @tagno", conn);
```

```
cmd.Parameters.Add("@tagno", SqlDbType.NVarChar).Value =
tagno;

SqlCeDataReader rdr = cmd.ExecuteReader();
if (rdr.Read() == false)
{
    MessageBox.Show("Kart kayıtlı değil.");
}
else
{
    adsoyadlbl1.Text = (string)rdr["adsoyad"];
    int bakiye = (int)rdr["bakiye"];
    baknowlbl1.Text = bakiye.ToString();
}
}
finally
{
    conn.Close();
}
kasardr.ClosePort();
}
catch
{
    MessageBox.Show("Okuyucu bulunamadı.");
}
}

private void guncellebtn1_Click(object sender, EventArgs e)
{
    if (tagnolbl1.Text.Equals("....."))
    {
        MessageBox.Show("Lütfen kart seçiniz.");
    }
    else
    {
        SqlCeConnection conn = new SqlCeConnection(dbpath);
        try
        {
            conn.Open();
            int bakiye = Convert.ToInt32(baktb1.Text);
            string tagno = tagnolbl1.Text;
            SqlCeCommand cmd = new SqlCeCommand("update musteritable
set bakiye = @bakiye where tagno = @tagno");
            cmd.Connection = conn;
            cmd.Parameters.Add("@bakiye", SqlDbType.Int).Value =
bakiye;
            cmd.Parameters.Add("@tagno", SqlDbType.NVarChar).Value =
tagno;

            int temp = cmd.ExecuteNonQuery();
            if (temp == 0)
                MessageBox.Show("Kart kayıtlı değil.");
        }
        catch
        {
            MessageBox.Show("Eksik ya da yanlış bilgi girdiniz, lütfen
tekrar deneyiniz.");
        }
        finally
        {
            conn.Close();
        }
    }
}
```

```
    }  
  }  
  
  private void iptalbtn1_Click(object sender, EventArgs e)  
  {  
    tagnolbl1.Text = ".....";  
    adsoyadlbl1.Text = ".....";  
    baknowlbl1.Text = ".....";  
    baktb1.Text = null;  
  }  
  
  private void kartsecbtn2_Click(object sender, EventArgs e)  
  {  
    try  
    {  
      string tagno = tagnoal();  
      tagnolbl2.Text = tagno;  
      kasardr.ClosePort();  
    }  
    catch  
    {  
      MessageBox.Show("Okuyucu bulunamadı.");  
    }  
  }  
  
  private void kaydetbtn2_Click(object sender, EventArgs e)  
  {  
    if (tagnolbl2.Text.Equals("....."))  
    {  
      MessageBox.Show("Lütfen kart seçiniz.");  
    }  
    else  
    {  
      SqlConnection conn = new SqlConnection(dbpath);  
      try  
      {  
        conn.Open();  
        int dogumyili = Convert.ToInt32(dogumyilib2.Text);  
        int bakiye = Convert.ToInt32(baktb2.Text);  
        string tagno = tagnolbl2.Text;  
        string adsoyad = adsoyadtb2.Text;  
        string tcno = tcnotb2.Text;  
        SqlCommand cmd1 = new SqlCommand("select tcno from  
musteritable where tcno = @tcno", conn);  
        cmd1.Parameters.Add("@tcno", SqlDbType.NVarChar).Value =  
tcno;  
  
        SqlDataReader rdr = cmd1.ExecuteReader();  
        if (rdr.Read() == false)  
        {  
          SqlCommand cmd2 = new SqlCommand("insert into  
musteritable (tagno, adsoyad, dogumyili, bakiye, tcno) values (@tagno,  
@adsoyad, @dogumyili, @bakiye, @tcno)", conn);  
          cmd2.Parameters.Add("@tagno",  
SqlDbType.NVarChar).Value = tagno;  
          cmd2.Parameters.Add("@adsoyad",  
SqlDbType.NVarChar).Value = adsoyad;  
          cmd2.Parameters.Add("@tcno", SqlDbType.NVarChar).Value  
= tcno;  
          cmd2.Parameters.Add("@dogumyili", SqlDbType.Int).Value  
= dogumyili;
```

```
        cmd2.Parameters.Add("@bakiye", SqlDbType.Int).Value =
bakiye;
        cmd2.ExecuteNonQuery();
    }
    else
    {
        MessageBox.Show("Kişi zaten kayıtlı.");
    }
}
catch
{
    MessageBox.Show("Eksik ya da yanlış bilgi girdiniz, lütfen
tekrar deneyiniz.");
}
finally
{
    conn.Close();
}
}

private void iptalbtn2_Click(object sender, EventArgs e)
{
    tagno1b12.Text = ".....";
    baktb2.Text = null;
    adsoyadtb2.Text = null;
    tcnotb2.Text = null;
    dogumyilitb2.Text = null;
}

private void kayitarabtn3_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(dbpath);
    try
    {
        conn.Open();
        string tcno = tcnotb3.Text;
        SqlCommand cmd = new SqlCommand("select adsoyad from
musteritable where tcno = @tcno", conn);
        cmd.Parameters.Add("@tcno", SqlDbType.NVarChar).Value = tcno;
        SqlDataReader rdr = cmd.ExecuteReader();
        if (rdr.Read() == false)
        {
            MessageBox.Show("Kişi kayıtlı değil.");
        }
        else
        {
            adsoyad1b13.Text = (string)rdr["adsoyad"];
            MessageBox.Show("Kayıt bulundu, lütfen kart seçiniz.");
        }
    }
    finally
    {
        conn.Close();
    }
}

private void kartsecbtn3_Click(object sender, EventArgs e)
{
    if (adsoyad1b13.Text.Equals("....."))
```



```
{
    MessageBox.Show("Önce T.C. kimlik numarası ile kaydı
aramalısınız.");
}
else
{
    try
    {
        string tagno = tagnoal();
        tagno1b13.Text = tagno;
        kasardr.ClosePort();
    }
    catch
    {
        MessageBox.Show("Okuyucu bulunamadı.");
    }
}
}

private void yenilebtn3_Click(object sender, EventArgs e)
{
    if (adsoyad1b13.Text.Equals("....."))
    {
        MessageBox.Show("Önce T.C. kimlik numarası ile kaydı
aramalısınız.");
    }
    else
    {
        if (tagno1b13.Text.Equals("....."))
        {
            MessageBox.Show("Lütfen kart seçiniz.");
        }
        else
        {
            SqlConnection conn = new SqlConnection(dbpath);
            try
            {
                conn.Open();
                int bakiye = Convert.ToInt32(baktb3.Text);
                string tcno = tcnotb3.Text;
                string tagno = tagno1b13.Text;
                SqlCommand cmd1 = new SqlCommand("select bakiye
from musteritable where tagno = @tagno", conn);
                cmd1.Parameters.Add("@tagno",
SqlDbType.NVarChar).Value = tagno;
                SqlDataReader rdr = cmd1.ExecuteReader();
                if (rdr.Read() == false)
                {
                    SqlCommand cmd2 = new SqlCommand("update
musteritable set bakiye = @bakiye, tagno = @tagno where tcno = @tcno");
                    cmd2.Connection = conn;
                    cmd2.Parameters.Add("@tcno",
SqlDbType.NVarChar).Value = tcno;
                    cmd2.Parameters.Add("@tagno",
SqlDbType.NVarChar).Value = tagno;
                    cmd2.Parameters.Add("@bakiye",
SqlDbType.Int).Value = bakiye;
                    cmd2.ExecuteNonQuery();
                }
            }
            else

```

```
        {
            MessageBox.Show("Bu kart kayıtlı, başka kart
kullanınız.");
        }
    }
    catch
    {
        MessageBox.Show("Eksik ya da yanlış bilgi girdiniz,
lütfen tekrar deneyiniz.");
    }
    finally
    {
        conn.Close();
    }
}
}

private void iptalbtn3_Click(object sender, EventArgs e)
{
    adsoyadlbl3.Text = ".....";
    tagnolbl3.Text = ".....";
    tcnotb3.Text = null;
    baktb3.Text = null;
}

private void kartsecbtn4_Click(object sender, EventArgs e)
{
    try
    {
        string tagno = tagnoal();
        tagnolbl4.Text = tagno;

        SqlConnection conn = new SqlConnection(dbpath);
        try
        {
            conn.Open();
            SqlCommand cmd = new SqlCommand("select adsoyad,
bakiye from musteritable where tagno = @tagno", conn);
            cmd.Parameters.Add("@tagno", SqlDbType.NVarChar).Value =
tagno;

            SqlDataReader rdr = cmd.ExecuteReader();
            if (rdr.Read() == false)
            {
                MessageBox.Show("Kişi kayıtlı değil.");
            }
            else
            {
                adsoyadlbl4.Text = (string)rdr["adsoyad"];
                int bakiye = (int)rdr["bakiye"];
                baknowlbl4.Text = bakiye.ToString();
                MessageBox.Show("Kayıt          bulundu,          çıkış
yapabilirsiniz.");
            }
        }
        finally
        {
            conn.Close();
        }
        kasardr.ClosePort();
    }
}
```

```
    }
    catch
    {
        MessageBox.Show("Okuyucu bulunamadı.");
    }
}

private void cikisbtn4_Click(object sender, EventArgs e)
{
    if (tagno1b14.Text.Equals("....."))
    {
        MessageBox.Show("Lütfen kart seçiniz.");
    }
    else
    {
        SqlConnection conn = new SqlConnection(dbpath);
        try
        {
            conn.Open();
            string tagno = tagno1b14.Text;
            SqlCommand cmd = new SqlCommand("update musteritable
set tagno = '0', bakiye = '0' where tagno = @tagno");
            cmd.Connection = conn;
            cmd.Parameters.Add("@tagno", SqlDbType.NVarChar).Value =
tagno;
            cmd.ExecuteNonQuery();
        }
        finally
        {
            conn.Close();
        }
    }
}

private void iptalbtn4_Click(object sender, EventArgs e)
{
    baknow1b14.Text = ".....";
    adsoyad1b14.Text = ".....";
    tagno1b14.Text = ".....";
}

private void kayitarabtn5_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(dbpath);
    try
    {
        conn.Open();
        string tcno = tcnotb5.Text;
        SqlCommand cmd = new SqlCommand("select adsoyad from
musteritable where tcno = @tcno", conn);
        cmd.Parameters.Add("@tcno", SqlDbType.NVarChar).Value = tcno;
        SqlDataReader rdr = cmd.ExecuteReader();
        if (rdr.Read() == false)
        {
            MessageBox.Show("Kişi kayıtlı değil.");
        }
        else
        {
            adsoyad1b15.Text = (string)rdr["adsoyad"];
            MessageBox.Show("Kayıt bulundu, lütfen kart seçiniz.");
        }
    }
}
```

```
    }
  }
  finally
  {
    conn.Close();
  }
}

private void kartsecbtn5_Click(object sender, EventArgs e)
{
  if (adsoyadlbl5.Text.Equals("....."))
  {
    MessageBox.Show("Önce T.C. kimlik numarası ile kaydı
aramalısınız.");
  }
  else
  {
    try
    {
      string tagno = tagnoal();
      tagnolbl5.Text = tagno;
      kasardr.ClosePort();
    }
    catch
    {
      MessageBox.Show("Okuyucu bulunamadı.");
    }
  }
}

private void yenikartbtn5_Click(object sender, EventArgs e)
{
  if (adsoyadlbl5.Text.Equals("....."))
  {
    MessageBox.Show("Önce T.C. kimlik numarası ile kaydı
aramalısınız.");
  }
  else
  {
    if (tagnolbl5.Text.Equals("....."))
    {
      MessageBox.Show("Lütfen kart seçiniz.");
    }
    else
    {
      SqlConnection conn = new SqlConnection(dbpath);
      try
      {
        conn.Open();
        string tcno = tcnotb5.Text;
        string tagno = tagnolbl5.Text;
        SqlCommand cmd1 = new SqlCommand("select bakiye
from musteritable where tagno = @tagno", conn);
        cmd1.Parameters.Add("@tagno",
SqlConnectionType.NVarChar).Value = tagno;
        SqlDataReader rdr = cmd1.ExecuteReader();
        if (rdr.Read() == false)
        {
          SqlCommand cmd2 = new SqlCommand("update
musteritable set tagno = @tagno where tcno = @tcno");

```

```
        cmd2.Connection = conn;
        cmd2.Parameters.Add("@tcno",
SqlDbType.NVarChar).Value = tcno;
        cmd2.Parameters.Add("@tagno",
SqlDbType.NVarChar).Value = tagno;
        cmd2.ExecuteNonQuery();
    }
    else
    {
        MessageBox.Show("Bu kart kayıtlı, başka kart
kullanınız.");
    }
}
finally
{
    conn.Close();
}
}
}

private void iptalbtn5_Click(object sender, EventArgs e)
{
    adsoyadlbl5.Text = ".....";
    tcnotb5.Text = "";
    tagnolbl5.Text = ".....";
}

private void goruntulebtn6_Click(object sender, EventArgs e)
{
    if (guncb6.Text.Equals("") || saatcb6.Text.Equals(""))
    {
        MessageBox.Show("Lütfen yer, gün ve saat seçiniz.");
    }
    else
    {
        SqlConnection conn = new SqlConnection(dbpath);
        if (barrbtn6.Checked)
        {
            if (guncb6.Text.Equals("Tüm Günler") &&
saatcb6.Text.Equals("Tüm Saatler")) //Toplam okunma sayısı.
            {
                try
                {
                    conn.Open();
                    SqlCommand cmd1 = new SqlCommand("select
count(*) as total from bartable", conn);
                    SqlDataReader rdr1 = cmd1.ExecuteReader();
                    rdr1.Read();
                    int count = (int)rdr1["total"];
                    grslbl6.Text = count.ToString();
                }
                finally
                {
                    conn.Close();
                }
            }
            else if (guncb6.Text.Equals("Tüm Günler") &&
!saatcb6.Text.Equals("Tüm Saatler")) //Seçilen saatlerdeki toplam okunma
sayısı.
```

```
{
    try
    {
        conn.Open();
        string saat = saatcb6.Text;
        SqlCommand cmd1 = new SqlCommand("select
count(saat) as total from bartable where saat = @saat", conn);
        cmd1.Parameters.Add("@saat",
SqlDbType.NVarChar).Value = saat;
        SqlDataReader rdr1 = cmd1.ExecuteReader();
        rdr1.Read();
        int count = (int)rdr1["total"];
        grslbl6.Text = count.ToString();
    }
    finally
    {
        conn.Close();
    }
}
else if (!guncb6.Text.Equals("Tüm Günler") &&
saatcb6.Text.Equals("Tüm Saatler")) //Seçilen gündeki toplam okunma sayısı.
{
    try
    {
        conn.Open();
        string gun = guncb6.Text;
        SqlCommand cmd1 = new SqlCommand("select
count(gun) as total from bartable where gun = @gun", conn);
        cmd1.Parameters.Add("@gun",
SqlDbType.NVarChar).Value = gun;
        SqlDataReader rdr1 = cmd1.ExecuteReader();
        rdr1.Read();
        int count = (int)rdr1["total"];
        grslbl6.Text = count.ToString();
    }
    finally
    {
        conn.Close();
    }
}
else //Seçilen günün seçilen saatlerdeki okunma sayısı.
{
    try
    {
        conn.Open();
        string gun = guncb6.Text;
        string saat = saatcb6.Text;
        SqlCommand cmd1 = new SqlCommand("select
count(date) as total from bartable where gun = @gun AND saat = @saat", conn);
        cmd1.Parameters.Add("@gun",
SqlDbType.NVarChar).Value = gun;
        cmd1.Parameters.Add("@saat",
SqlDbType.NVarChar).Value = saat;
        SqlDataReader rdr1 = cmd1.ExecuteReader();
        rdr1.Read();
        int count = (int)rdr1["total"];
        grslbl6.Text = count.ToString();
    }
    finally
    {

```

```
        conn.Close();
    }
}
else if (havuzrbtn6.Checked)
{
    if (guncb6.Text.Equals("Tüm Günler") &&
saatcb6.Text.Equals("Tüm Saatler")) //Toplam okunma sayısı.
    {
        try
        {
            conn.Open();
            SqlCommand cmd1 = new SqlCommand("select
count(*) as total from havuztable", conn);
            SqlDataReader rdr1 = cmd1.ExecuteReader();
            rdr1.Read();
            int count = (int)rdr1["total"];
            grslbl6.Text = count.ToString();
        }
        finally
        {
            conn.Close();
        }
    }
    else if (guncb6.Text.Equals("Tüm Günler") &&
!saatcb6.Text.Equals("Tüm Saatler")) //Seçilen saatlerdeki toplam okunma
sayısı.
    {
        try
        {
            conn.Open();
            string saat = saatcb6.Text;
            SqlCommand cmd1 = new SqlCommand("select
count(saat) as total from havuztable where saat = @saat", conn);
            cmd1.Parameters.Add("@saat",
SqlDbType.NVarChar).Value = saat;
            SqlDataReader rdr1 = cmd1.ExecuteReader();
            rdr1.Read();
            int count = (int)rdr1["total"];
            grslbl6.Text = count.ToString();
        }
        finally
        {
            conn.Close();
        }
    }
    else if (!guncb6.Text.Equals("Tüm Günler") &&
saatcb6.Text.Equals("Tüm Saatler")) //Seçilen gündeki toplam okunma sayısı.
    {
        try
        {
            conn.Open();
            string gun = guncb6.Text;
            SqlCommand cmd1 = new SqlCommand("select
count(gun) as total from havuztable where gun = @gun", conn);
            cmd1.Parameters.Add("@gun",
SqlDbType.NVarChar).Value = gun;
            SqlDataReader rdr1 = cmd1.ExecuteReader();
            rdr1.Read();
            int count = (int)rdr1["total"];
        }
    }
}
```

```
        grslbl6.Text = count.ToString();
    }
    finally
    {
        conn.Close();
    }
}
else //Seçilen günün seçilen saatlerdeki okunma sayısı.
{
    try
    {
        conn.Open();
        string gun = guncb6.Text;
        string saat = saatcb6.Text;
        SqlCommand cmd1 = new SqlCommand("select
count(date) as total from havuztable where gun = @gun AND saat = @saat",
conn);
        cmd1.Parameters.Add("@gun",
SqlDbType.NVarChar).Value = gun;
        cmd1.Parameters.Add("@saat",
SqlDbType.NVarChar).Value = saat;
        SqlDataReader rdr1 = cmd1.ExecuteReader();
        rdr1.Read();
        int count = (int)rdr1["total"];
        grslbl6.Text = count.ToString();
    }
    finally
    {
        conn.Close();
    }
}
}
else
{
    MessageBox.Show("Lütfen yer, gün ve saat seçiniz.");
}
}
}

private void temizlebtn6_Click(object sender, EventArgs e)
{
    guncb6.Text = "";
    saatcb6.Text = "";
    barrbtn6.Checked = false;
    havuzrbtn6.Checked = false;
    grslbl6.Text = ".....";
}
}
}
```



## EK 4. PIC C Uygulaması Kaynak Kodları

```
#include <16f877A.h>

#fuses XT,NOWDT,NOPROTECT,NOBROWNOUT,NOLVP,NOPT,NOVRT,NODEBUG,NOCPD

#use delay (clock=4000000)

#use rs232 (baud=9600, xmit=pin_c6, rcv=pin_c7, parity=N, stop=1, bits=8)

char received;

#int_rda
void serial_interrupt()
{
    disable_interrupts(int_rda);
    received = getc();
    if(received == 'y')
    {
        output_high(pin_c5); //green led
        delay_ms(600);
        output_low(pin_c5);
    }
    else
    {
        output_high(pin_c4); //red led
        delay_ms(600);
        output_low(pin_c4);
    }
}

void main()
{
    setup_psp(PSP_DISABLED);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_OFF);
    setup_CCP1(CCP_OFF);
    setup_CCP2(CCP_OFF);

    output_low(pin_c4);
    output_low(pin_c5);

    enable_interrupts(GLOBAL);
    while(1)
    {
        enable_interrupts(int_rda);
    }
}
```

## ÖZGEÇMİŞ

**Adı Soyadı:** Mehmet Mütin TÜRK

**Doğum Tarihi:** 03/08/1984

**Öğrenim Durumu:**

| Derece        | Bölüm/Program                    | Üniversite           | Yıl       |
|---------------|----------------------------------|----------------------|-----------|
| Lise          | Fen Bilimleri                    | Kabataş Erkek Lisesi | 1999-2002 |
| Lisans        | Elektrik-Elektronik Mühendisliği | Mersin Üniversitesi  | 2005-2009 |
| Yüksek Lisans | Elektrik-Elektronik Mühendisliği | Mersin Üniversitesi  | 2010-     |