

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

103990

**İTÜKART
YÖNETİM OTOMASYONU**

**T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ**

**YÜKSEK LİSANS TEZİ
Müh. Ece ÇİNER
(504971605)**

103980

**Tezin Enstitüye Verildiği Tarih : 11 Haziran 2001
Tezin Savunulduğu Tarih : 29 Haziran 2001**

**Tez Danışmanı : Doç.Dr. A. Coşkun SÖNMEZ
Diğer Jüri Üyeleri Prof.Dr. Bülent ÖRENCİK
Yrd.Doç.Dr. Selçuk PAKER**

[Signatures]

HAZİRAN 2001

ÖNSÖZ

Akıllı kartlar, barındırdıkları bellek ve mikroişlemci sayesinde bilgi saklayıp işleyebilirler. Fiziksel olarak kredi kartlarına oldukça benzeyen bu plastik kartlar, özellikle sundukları yüksek güvenlik ve taşınabilir, kolay kullanılabilir olmaları nedeniyle günümüzde oldukça geniş uygulama alanı bulmaktadırlar.

Bu tez çalışmasında, akıllı kartların Türkiye'deki kullanımının artmasına katkıda bulunmak amacıyla örnek bir kampüs kartı uygulaması olarak geliştirilen İTÜKart projesinin yönetim işleri bölümü hazırlanmıştır.

Bütün bu zorlu çalışma boyunca yardımlarını esirgemeyen değerli danışmanım Sayın Doç. Dr. A. Coşkun SÖNMEZ'e, uygulamanın geliştirilmesi aşamasında destek veren Sayın Güney SEYHAN şahsında KoçSistem Bilgi ve İletişim Hizmetleri A.Ş.'ye, Burak BESTEL'e, Tarkan Alptekin DURMUŞ'a ve her zaman yanımda olan aileme katkılarından dolayı teşekkür ederim.

Haziran, 2001

Ece ÇİNER

İÇİNDEKİLER

KISALTMALAR	v
TABLO LİSTESİ	vi
ŞEKİL LİSTESİ	vii
SEMBOL LİSTESİ	ix
İTÜKART YÖNETİM OTOMASYONU	x
İTÜKART ADMINISTRATION AUTOMATION	xii
1. GİRİŞ	1
2. AKILLI KARTLAR	3
2.1. Akıllı Kartların Tarihçesi	3
2.2. Akıllı Kart Tipleri	4
2.2.1. Kart ile Terminal Arasındaki İletişim Şekline Göre Akıllı Kartlar	4
2.2.2. İçeriklerine Göre Akıllı Kartlar	5
2.3. Veri Yapısı	6
2.4. Güvenlik	7
2.5. Akıllı Kartların Uygulama Alanları	8
3. İTÜKART UYGULAMASI	11
3.1. Yazılım ve Donanım Gereksinimleri	13
3.2. Kullanılan Teknolojiler ve Diller	14
3.3. İşlevler	14
3.4. Veritabanı	17
3.4.1. Tablolar	17
3.4.2. Bağımlılıklar	18
3.5. Güvenlik	18
3.6. Modüller	20
3.6.1. GemPlus410 Arayüz Kontrolü	20
3.6.2. SCOjects ActiveX DLL'i	28
3.6.3. ITUEncrypt ActiveX DLL'i	29
3.6.4. Ekranlar	30
SONUÇ	93
KAYNAKLAR	96
EKLER	98
Ek.A. GemClub Akıllı Kartı	98
Ek.B. Veritabanı Yapısı	132
Ek.C. Kontaklı Kartlarda Standart İletişim Protokolleri	137



KISALTMALAR

ADO	: ActiveX Data Object
APDU	: Application Protocol Data Unit
BBS	: Blok Bekleme Süresi
BKZ	: Blok Koruma Zamanı
CAD	: Card Accepting Device
CEN	: Comité Européen de Normalisation
COM	: Component Object Model
CPU	: Central Processing Unit
CRC	: Cyclic redundancy check
CTC	: Card Transaction Counter
DES	: Data Encryption Standard
EEPROM	: Electrically Erasable Programmable Read Only Memory
EPROM	: Electrically Programmable Read Only Memory
ETSI	: European Telecommunications Standards Institute
ETU	: Elementary Time Unit
IC	: Integrated Circuit
ICC	: Integrated Circuit Card
IEC	: International Electrotechnical Commission
IFD	: Interface Device
IIS	: Internet Information Server
KBS	: Karakter Bekleme Süresi
LRC	: Longitudinal redundancy check
MAC	: Message Authentication Code
MİB	: Merkezi İşlem Birimi
OSI	: Open System Interconnection
OTP	: One-time Programmable
PIN	: Personal Identification Number
PKB	: Protokol Kontrol Byte'ı
PROM	: Programmable Read Only Memory
PSE	: Payment System Environment
PVC	: Poli vinil klorür
RAM	: Random Access Memory
ROM	: Read Only Memory
SFI	: System File Identifier
SIM	: Subscriber Identification Number
SIO	: Serial Input Output
TLV	: Tag Length Value
TPDU	: Transport Protocol Data Unit

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo A.1. Dosya Tipleri.....	100
Tablo A.2. Sistem Dosyası Yapısı.....	102
Tablo A.3. Sayaç Dosyası Yapı Alanı.....	103
Tablo A.4. Sayaç Dosyası Yapısı.....	103
Tablo A.5. Sayaç Dosyası Seçimlik Özellikler.....	104
Tablo A.6. GemClub Tarih Dönüşümü.....	105
Tablo A.7. Kural Dosyası Yapısı.....	106
Tablo A.8. Kural Dosyasına Başka Makrokomut Eklenmesi.....	106
Tablo A.9. Makrokomut Yapısı.....	107
Tablo A.10. Gizli Kod Dosyasının Yapısı.....	108
Tablo A.11. Gizli Anahtar Dosyasının Yapısı.....	109
Tablo A.12. Kayıt Dosyasının Yapısı.....	111
Tablo A.13. EMV_DIR Dosyasının Yapısı.....	112
Tablo A.14. Sistem Dosyası Başlangıç Parametreleri.....	114
Tablo A.15. Anahtar Dosyası Parametreleri.....	114
Tablo A.16. Erişim Koşulları.....	114
Tablo A.17. EEPROM Büyüklüğü.....	115
Tablo A.18. Erişim koşulları için kullanılan fonksiyonlar.....	117
Tablo A.19. Erişim koşullarında yer alan kodlar için kurallar.....	118
Tablo A.20. Dolaylı Erişim Koşuluna Sahip Veri Elemanları.....	118
Tablo A.21. Ödül işlemlerinde kullanılan MAC değerleri.....	125
Tablo A.22. İndirim işlemlerinde kullanılan MAC değerleri.....	126
Tablo A.23. Kural Kullanma işlemlerinde kullanılan MAC değerleri.....	127
Tablo A.24. Diğer işlemlerde kullanılan MAC değerleri.....	128
Tablo A.25. Önceden Tanımlanmış Etiket Değerleri.....	129
Tablo C.1. PKB Kodlama Tablosu.....	148

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1	Kontaklı Kartlar.....	4
Şekil 3.1	İTÜKart Genel Sistem Mimarisi.....	12
Şekil 3.2	Yazılım Fonksiyonlarının Genel Giriş-Çıkış Şeması.....	16
Şekil 3.3	ITU Veritabanı Yapısı.....	19
Şekil 3.4	İTÜKart İçin Kullanılan Dosya Yapısı.....	21
Şekil 3.5	Yetkili-Rol-Ekran Yapısı.....	31
Şekil 3.6	Giriş Ekranı.....	32
Şekil 3.7	Ana Ekran.....	34
Şekil 3.8	Kullanıcı Tanımları – Ana Ekran.....	37
Şekil 3.9	Kimlik Bilgileri Ekranı.....	41
Şekil 3.10	Kampüs Tanımları.....	44
Şekil 3.11	Bina Tanımları.....	46
Şekil 3.12	Sınıf Tanımları.....	49
Şekil 3.13	Fakülte Tanımları.....	52
Şekil 3.14	Enstitü Tanımları.....	55
Şekil 3.15	Bölüm Tanımları.....	58
Şekil 3.16	Ana Bilim Dalı Tanımları.....	62
Şekil 3.17	Ders Tanımları.....	65
Şekil 3.18	Ön Koşul Tanımlama.....	70
Şekil 3.19	Ders Programı Tanımları.....	75
Şekil 3.20	Not Girişi.....	80
Şekil 3.21	Parametre Girişi.....	84
Şekil 3.22	Rol Tanımları.....	87
Şekil 3.23	Ekran Yetkileri.....	90
Şekil A.1	Dosya Yapısı.....	99
Şekil A.2	GemClub Tarih Gösterimi.....	105
Şekil A.3	Değişken Uzunluklu Kayıtlar İçeren Doğrusal Dosya.....	110
Şekil A.4	Başlangıçtaki Dosya Yapısı.....	113
Şekil A.5	Seri Numarası.....	115
Şekil A.6	Erişim Koşulu Yapısı.....	117
Şekil A.7	Parametre Güncelleme İçin Güvenli Mesajlaşma.....	120
Şekil A.8	İşlem Sınama.....	122
Şekil A.9	3DES Şifreleme / Şifre Çözme Algoritması.....	124
Şekil A.10	MAC Hesabı.	125
Şekil C.1	Karakter Çerçevesi	137
Şekil C.2	Albaştan Yanıtı.....	139
Şekil C.3	Başlangıç Karakteri.....	140

Şekil C.4	Zorunlu Olmayan Arabirim Byte'ları.....	140
Şekil C.5	TA2 Arabirim Byte'ı.....	142
Şekil C.6	PTR İsteği.....	144
Şekil C.7	T=0 Protokolünde Veri Akışı.....	145
Şekil C.8	Blok Penceresi.....	146
Şekil C.9	T=1 Protokolü ve Normal I İletişimi Gönderimi.....	149
Şekil C.10	I Blok İletişiminde Zincirleme.....	149
Şekil C.11	I Blok Alımında Hata	149
Şekil C.12	I Blok Yanıt Zincirinde Hata	150



SEMBOL LİSTESİ

- V_{CC}** : Kart besleme gerilimi
 V_{SS} : V_{CC} besleme geriliminin ölçümünde kullanılan toprak seviyesi gerilimi
 V_{PP} : Karttaki EPROM belleğin programlanması sırasında kullanılan yüksek gerilim.



İTÜKART YÖNETİM OTOMASYONU

ÖZET

Gövdesinde yer alan tümleşik devre sayesinde veri depolayıp işleyebilen ve fiziksel özellik bakımından kredi kartlarını andıran plastik kartlara genel olarak 'Akıllı Kart' adı verilir. Bu kartlar, manyetik benzerlerine göre yaklaşık bin kat daha fazla veri saklayabilmektedirler. İçerdikleri ileri seviyedeki kontrol mekanizmaları sayesinde daha güvenilirlerdir. Bunun yanı sıra aynı kart birden fazla uygulama içinde kullanılabilmesinden benzerlerine göre daha fazla tercih edilmektedirler. Dünyada hızla yaygınlaşmalarının altında yatan sebeplerden bir tanesi de budur.

Akıllı kartlar, günümüzde kimlik tanıma, bankacılık, elektronik cüzdan, ulaşım gibi pek çok alanda yaygın olarak kullanılmaktadır. İçerdikleri kırmıklarda değişik kullanım amaçları için, değişik tipte bilgiler depolanabilmektedir. Alışverişte kullanılmak üzere para taşınabileceği gibi, sağlık ile ilgili bilgiler ve uygulamalar da güvenli bir şekilde tutulabilir.

Ülkemizde elektronik cüzdan ve ulaşım alanlarında kullanılmaya başlanan ve gittikçe yaygınlaşan akıllı kart uygulamalarına bir örnek oluşturması amacıyla tez kapsamında İTÜKart projesi hazırlanmıştır. Bu tez çalışmasında, üniversitelerdeki öğrenci, öğretim üyesi ve diğer çalışanlar için kimlik kartı ve elektronik cüzdan işlemlerini gerçekleştiren sistemin bir prototipi oluşturulmuştur. Proje iki alt bölümden oluşmaktadır: İTÜKart Yönetim Otomasyonu ve İTÜKart Kullanıcı Otomasyonu.

İTÜKart Yönetim Otomasyonu'nda, İTÜKart sisteminin çalışması için gerekli olan ve sadece yetkili kişilerce yapılabilecek tanımlamalar yer almaktadır. İTÜKart uygulamasının birimlerine erişim için İTÜKart sahibi olmak şarttır. Kullanıcıların kendi İTÜKart'ları ile giriş yaptığı bu program üzerinden girilen

- Parametre tanımları,
- Kampüs-bina-sınıf tanımları,
- Bölüm-ana bilim dalı-fakülte-enstitü tanımları,
- Ders-ders programı-sınav programı,
- Kullanıcı tanımları

gibi veriler İTÜKart Kullanıcı Otomasyonu ile ortak kullanılan veritabanında tutulur [1].

İTÜKart Kullanıcı Otomasyonu'nda ise, kullanıcıların web ortamında, üzerinde kart okuyucu bulunan herhangi bir makinedeki tarayıcı ile erişebileceği ders, randevu ve ödeme işlemleriyle ilgili ekranlar bulunmaktadır [1].

İTÜKart projesi bünyesinde gerçekleştirilen her iki otomasyon programında kullanılmak üzere, akıllı kart ile kart okuyucunun iletişimini sağlayan bir arabirim yazılımı gerçekleştirilmiştir. Bu yazılımda yer alan fonksiyonlar sayesinde karttaki veriyi okumak, karta veri yazmak ve erişim kontrolleri uygulamak mümkündür.

İTÜKart projesi için, kullanılan GemClub kartı üzerinde kullanıcı numarası ve kullanıcının o an sahip olduğu kredi tutarı saklanmaktadır. Veri güvenliği, kart tarafından sunulan PIN (Personal Identification Number) ile sağlanmaktadır. Kartta verilerin saklandığı kayıt dosyasının okunması işlemi PIN, güncellenmesi işlemi ise hem PIN, hem de dosyanın gizli anahtar (secret key) değeri ile korunmaktadır. Ayrıca karttaki PIN'in değiştirilmesi için de buna ait farklı bir gizli anahtar değerinin belirtilmesi gerekmektedir.

Uygulamada karttaki güvenlik mekanizmalarının yanısıra, kart sahibine ait olup veritabanında tutulan bilgilerin güvenliği de şifrelenmiş olarak saklanan kullanıcı parolaları ile sağlanmaktadır. Böylece uygulamadaki iki katmanlı koruma mekanizması ile veri güvenliğinin artırılması amaçlanmıştır.

Prototip olarak hazırlanan ve modüler bir yapıya sahip olan İTÜKart uygulamasında sadece benzer sistemlerin temel fonksiyonları ele alınmıştır. Bu yapı üzerinde, ileride ortaya çıkabilecek yeni ihtiyaçlar doğrultusunda kapsamın kolaylıkla genişletilmesi mümkündür.

İTÜKART ADMINISTRATION AUTOMATION

SUMMARY

Plastic cards, similar to credit cards which an embeded integrated circuit enabling it to store and process data are called ‘Smart Cards’. These cards can store approximately a thousand times more data compared to magnetic cards. Since the same card can be used in several different applications, smart cards are widely preferred tools for secure identification.

Smart cards are widely used in many areas like identification, banking, electronic purse and transportation. Different types of data, as well as applications, can be stored in a secure way in the chips for different purposes such as keeping credit for purchase or information about the cardholder’s background concerning health.

In Turkey, applications in electronic purse and transportation have started to show up in market, therefore smart card usage is expected to accelerate. The İTÜKart project is developed as a part of this thesis for providing an example to smart card applications. İTÜKart is a prototype for a university card to be used by students, faculty, and other academic personnel as an id card and an electronic purse. It is required to have an İTÜKart card to access the units in the system. The project consists of two parts: İTÜKart Administration Automation and İTÜKart User Automation.

With İTÜKart Administration Automation, authorized users can access the program using their İTÜKarts and can make and maintain the definitions required for the system like

- Parameter definitions,
- Campus-building-classroom definitions,
- Department-institution definitions,
- Lesson–schedule definitions,
- User definitions.

The data is stored on a common database also used by the user automation [1].

İTÜKart User Automation consists of screens, which are accessed through internet via terminals with smart card readers and internet browsers. With the help of these screens the students can carry out operations related with lessons, reservations and payment [1].

To provide a means of communication between the card and the reader an interface software is also developed during this thesis work. It is possible to read or update the data on the card by calling the methods of this interface.

In the İTÜKart application, the user id of the cardholder and the amount of credit that the cardholder has are stored on the GemClub card. The PIN mechanism of the card itself is used for guaranteeing data security. The read operation on the record file is protected by PIN whereas the update operation is protected by both the PIN and the secret key related to that record file. Furthermore a different secret key is required for updating the PIN on the card.

In the application context apart from the PIN mechanism, the security of the data in the database, which is directly related to the cardholder, is accomplished through the encrypted user password in the user definition. This two-layered protection mechanism used in the application ensures data security.

In the İTÜKart system, which is a prototype designed on a modular basis, only some basic functions of university cards are implemented. This structure can easily be expanded to meet future needs.

1. GİRİŞ

Hızla gelişen bilişim teknolojileri sayesinde günlük hayatımıza giren elektronik ticaret, elektronik bankacılık, elektronik bilet (Akbi, OGS, vb), elektronik kimlik gibi uygulamaların hayata geçmesi hız, kolay kullanım, yüksek erişilebilirlik ve düşük maliyet gibi bir çok avantajı beraberinde getirmiştir. Ancak faydalarının yanında bu tür sayısal uygulamalarda en büyük zorluk, kullanıcı kimliğinin güvenilir bir şekilde doğrulanmasında ortaya çıkmaktadır.

Yüksek güvenlik gerektiren sayısal uygulamalarda, kullanıcının gerçekten beyan ettiği kişi olup olmadığını ya da talep ettiği işlem için yetkili olup olmadığını anlamak amacıyla manyetik veya biometrik tanıma yöntemleri kullanılmaktadır. Kredi kartları düşük maliyetleri dolayısıyla oldukça yaygınlaşmış manyetik ürünlerdir. Fakat yeterli seviyede güvenlik sağlayamadıkları için sahtekarlıklara açıktırlar. Yüksek teknoloji gerektiren biometrik yöntemler ise oldukça güvenli bir altyapı üzerine inşa edilmişlerdir. Parmak izi, retina tarama gibi örnekleri olan biometrik uygulamalar çok yüksek maliyetlere sahip oldukları için tercih edilmemektedirler.

Manyetik ve biometrik yöntemlerin alternatifi olan akıllı kartlar, benzerleri arasında güvenlik/maliyet oranı açısından en uygun çözümü sunmaktadır. İçerdikleri bellek ve işlemci sayesinde sadece veri depolama ile sınırlı kalmayıp işlem yapma yeteneğine de sahip olan akıllı kartlar fiziksel olarak manyetik kartlara benzerler ve PIN mekanizması sayesinde kullanıcı doğrulama işlevini gerçekleştirirler. Akıllı kartlarda hem depolanan veri, hem kartı kullanmak için gerekli olan PIN, hem de okuyucu ile kart arasındaki iletişim şifrelenerek sahtekarlık olasılığı neredeyse sıfıra indirilir. Akıllı kartlar, düşük maliyetleri nedeniyle yüksek güvenlik sağlayan alternatif yöntemlere göre daha yaygın olarak kullanılmaktadırlar.

Dünyada hızla yaygınlaşmasına rağmen Türkiye’de çok fazla örneği bulunmayan kampüs kartları tez çalışması için konu olarak seçilmiştir. Hazırlanan İTÜKart Sistemi ile elektronik kimlik, elektronik cüzdan, öğrenci durum takibi, akademik takvimin ders programı detayına kadar belirlenebilmesi, sınıf/laboratuvar rezervasyonu gibi işlemlerin gerçekleştirilebildiği bir sistemin prototipi gerçekleştirilmiştir.

İTÜKart uygulaması iki alt bölümden oluşmaktadır. Bu yazılımlar ortak veritabanı üzerinde çalışmakta ve kart okuma-yazma işlemlerini aynı arabirim aracılığı ile gerçekleştirmektedirler.

İTÜKart Yönetim Otomasyonu : Enstitü görevlisi, öğretim üyesi gibi yetkili kişilerce gerçekleştirilecek, İTÜKart sisteminden beklenen işlevler için gerekli olan tanımlama ve giriş işlemlerinin yapıldığı programdır. Sadece sistemde tanımlanmış yetkili kişiler tarafından kullanılması amacına dayandığından istemci-sunucu mimarisi esas alınarak hazırlanmıştır.

İTÜKart Kullanıcı Otomasyonu : Başta öğrenciler olmak üzere İTÜKart kullanıcılarının 7 gün, 24 saat ulaşabildiği bu uygulama kullanılarak ders, randevu ve ödeme takibi ile ilişkili işlemler yapılabilmektedir [1].

Tez çalışması kapsamında gerçekleştirilen arabirim sayesinde akıllı kart ile kart okuyucu arasındaki iletişim sağlanmaktadır. ActiveX kontrolü (ActiveX control) olarak gerçekleştirilen arabirimde yer alan metodlar sayesinde PIN kontrolü yapmak, PIN değiştirmek, karttaki bilgileri okumak veya istenen bilgileri karta aktarmak mümkündür.

2. AKILLI KARTLAR

IC (Integrated Circuit) kartları veya ICC (Integrated Circuit Card) kartları olarak da adlandırılan akıllı kartlar, boyutları kredi kartı ile aynı olan, fakat tümleşik devreli kırımlar içeren plastik kartlardır. Akıllı kartların fiziksel özellikleri ISO 7816 – 1 standartı tarafından belirlenmiştir. Akıllı kartlarda yer alan tümleşik devreli kırımda sadece bellek bulunabileceği gibi, PIN kontrolünü yapan bir mantıksal devre veya karta karmaşık işlem yapabilme yeteneği katan bir mikroişlemci de içerebilir [2, 3].

2.1. Akıllı Kartların Tarihçesi

Akıllı kartların doğuşundaki en önemli kilometre taşı, mikroişlemcinin bulunmasıdır. 1971’de Ted HOFF’un, bilgisayarı ince bir silikon parçasının içine gömmeyi başarmasıyla mikroelettronik dünyasında bir devrim yaşanırken akıllı karta doğru ilk adım da atılmıştı. Bu kırımda sayesinde, daha büyük işlem güçleri çok daha küçük alanlar üzerinde gerçekleştirilmeye başlanmıştı. Bunun yanı sıra maliyetler ise giderek düşüyordu.

Intel ve mikrokırımda (microchip) endüstrisinin diğer önde gelen firmaları kırımda üzerindeki bilgisayarlar için bir pazar oluşturma çabası içindelerken, Rolan MORENO adlı bir Fransız kırımda bir kart üzerine yerleştirmişti. Bellek, mantıksal devre, mikroişlemci gibi bir bilgisayarın sahip olduğu tüm teknik özellikleri kırımda teknolojesi sayesinde bünyesinde barındırabilen akıllı kartlar, bütün elektronik sistemlerin en kolay taşınabiliri ve en pratiği olduğundan yaşamımızda ve modern toplumda vazgeçilmez bir öneme sahip hale gelmiştir.

Moreno’nun icadından 20 yıl sonra piyasada 20 milyondan fazla banka kartı, milyonlarca telefon kartı dolaşmaya başladığında bu kavram Fransa’da kabul gördü. Akıllı kartın geniş bir uygulama alanına ve yaygınlaşma potansiyeline sahip olduğunu

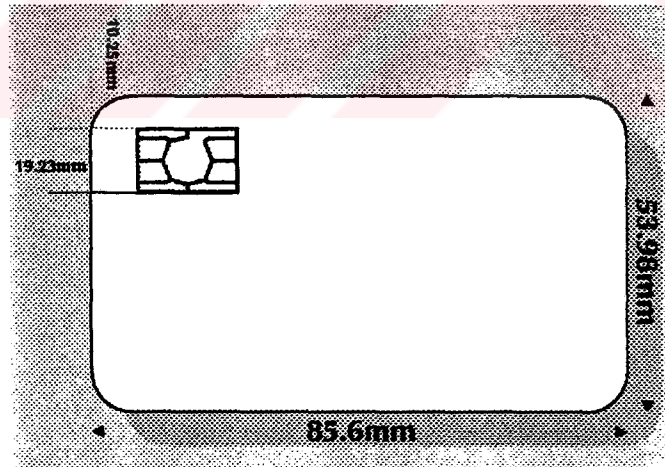
farkeden dünya apında 230'dan fazla firma bir araya gelerek akıllı kartların gelişimini sağlamak amacıyla Akıllı Kart Forumu'nu kurdular. [4, 5, 6].

2.2. Akıllı Kart Tipleri

Genel olarak akıllı kartlar, kart ile terminal arasındaki iletişim şekline ve barındırdıkları fiziksel bileşenlere göre iki başlıkta sınıflandırılabilirler.

2.2.1. Kart ile Terminal Arasındaki İletişim Şekline Göre Akıllı Kartlar

Kontaklı Kartlar: Şekil 2.1'de gösterilen kontaklı akıllı kartlar gerilim kaynağı, saat işareti ve veri alış verişi için giriş/çıkış hatları gibi dış dünya ile iletişimi gerektiren işaretleri kart üzerindeki kontaklar sayesinde doğrudan bağlantı ile alırlar. Kart üstündeki tümleşik devre, kartın yüzeyindeki kontak plakasına temas eder; akıllı kart, kart okuyucuya yerleştirildiğinde karttaki kontak plakası, okuyucudaki plaka ile örtüşerek devreyi tamamlar. Bu işlem için, kart üzerindeki plakanın ISO 7816-2 standartında belirtilen konumda olması gerekir [3, 7].



Şekil 2.1. Kontaklı kartlar

Kontaksız Kartlar: Kontaksız kartlarda terminal ile iletişim için elektriksel çiftleme (electrical coupling) kullanılır. Kontaksız kartların kart okuyucuya ortalama 3 cm gibi kısa bir mesafede tutulması yeterlidir. Güç, indüktif veya kapasitif çiftleme ile

terminalden karta iletilir. İletişimde kullanılan saat işareti dahili olarak üretilebilir. Terminal ve kart arasında ise veriler güç işareti module edilerek iletilebilir.

Daha uzak mesafelerde çalışabilen kontaklız akıllı kartlar radyo sinyalleri kullanırlar. Fakat gerekli enerjiyi radyo dalgalarında elde etmek zor olduđu için bu sistemlerde pil içeren jetonlar kullanılır. Jeton büyüklüğü ISO standartlarında belirtilen akıllı kart boyutlarına indirgenemeyeceğı için, radyo dalgalarının kullanımı akıllı kartlar için şu an pek mümkün değildir. Fakat gelişen teknoloji sayesinde, radyo sinyallerinden elde edilen gücün yeterli olacağı düşük güçlü kırmıkların kullanımı mümkün olacaktır. Ayrıca çok ince piller sayesinde böyle bir kırmık da plastik karta sığdırılabilecektir.

Kontaklı kartlarda kontak plakasının kirlenmesi ya da zarar görmesi gibi bir sorunlarla karşılaşılabilirdiğinden kontaklız kartlar daha avantajlı gözükmemektedir. Ayrıca kontaklız kartların okuyucuya yerleştirilmesi gerekmemektedir, sadece belirli bir mesafeye kadar yaklaştırılmaları yetmektedir. Bu sayede amaç dışı kötü niyetli kullanım potansiyeli azaltılmakta, daha hızlı bir iş akışı sağlanmaktadır [3].

Günümüzde tam olarak standartlaştırılamayan kontaklız akıllı kartların bazı dezavantajları da bulunmaktadır. Kontaklı kartlara göre bu kartların üretilmeleri oldukça yavaş ve pahalı bir işlemdir. İçerdikleri birbirine bağı birçok bileşen nedeniyle bükülmelere karşı daha az dayanıklıdırlar. Kabartma yöntemi ile üzerlerine yazı yazılması sırasında, içerdikleri bileşenler zedelendiğinden problemlerle karşılaşılmaktadır. Ayrıca kart ile okuyucu arasına konulan bir mekanizma sayesinde veriler dinlenebileceğı için daha az güvenlidirler. Ama teknolojinin ilerlemesi ile kontaklız kartlar da daha iyiye gitmektedir [3].

2.2.2. İçeriklerine Göre Akıllı Kartlar

İçeriklerine göre akıllı kartlar şu şekilde sınıflandırılabilir:

- Sadece bellek içerenler,
- Bellek ve güvenlik mekanizması içerenler,

- Bellek ve MİB içerenler.

Basit bir telefon kartı için 128 – 512 byte'lık küçük bir EEPROM belleğe sahip kart yeterli olurken daha karmaşık uygulamalar için ROM, EEPROM, RAM ve MİB gerekmektedir.

Akıllı kartlar barındırdıkları güvenlik mekanizması sayesinde alternatiflerinin önüne geçmiştir. Bu mekanizma ile belleğe, sadece gerekli yetkiye sahip kişilerin erişmesi sağlanır, böylece kötü niyetli kişilerin maddi kazanç sağlamaları önlenir. Bu iş için 64 bit veya daha uzun olan erişim kodları kullanılmaktadır. Güvenlik konusunda mikroişlemci içeren akıllı kartlar daha avantajlıdır; çünkü bu işlemci kullanımı sayesinde, daha karmaşık güvenlik algoritmalarının gerçekleştirilmesini mümkün olmaktadır [9].

Akıllı kart üzerinde yer alan mantıksal bir devre ile, terminalden (CAD - Card Accepting Device) gelen komutlar alınabilir, uygulama verilerinin değiş tokuşu gerçekleştirilebilir. Kullanılan akıllı kart MİB içeriyorsa terminalden gelen komutlara göre veri işleme ve karar alma yeteneği sayesinde, kart üzerinde birden fazla program çalıştırılabilir [9].

2.3. Veri Yapısı

Akıllı kartlar, üzerindeki veri yapısı biliniyorsa güvenlik koşulları karşılanmak şartı ile istenildiği gibi yazılıp okunabilir. Sadece belleği olan işlemcisiz kartlarda veri dosyası, programcının istediği gibi biçimlendirdiği düz bir dosya olarak düşünülebilir.

Akıllı kart üzerinde, işletim sistemi kontrolünde çalışan bir işlemci olması durumunda daha karmaşık bir veri yapısı söz konusudur. İşlemcili bir kartın veri alanı disk sürücününki gibi incelenebilir. ISO 7816-4'te standartları açıklanan yapıya göre, kart üzerindeki bellek alanı dosyalara bölümlenebilir ya da dizinler ile hiyerarşik bir yapıya dönüştürülebilir [10]. Birçok kırmık üreticisinin ISO 7816'ya uyan ürünler geliştirmesine rağmen sonuçta ortaya çıkan ürünler farklı veri yapıları nedeniyle birbiriyle uyuşmamaktadır.

Birkart üzerinde birden fazla uygulama olması durumunda uygulamaları yazan kişilerin veri alanının kullanımı hakkında görüş birliğine varmaları gerekmektedir. Bir uygulamanın verisi diğer uygulama tarafından okunabilir ya da bozulabilir. Bu durumları önlemek için uygulama geliştiriciler güvenliği arttırmak için işlemcili kartı tercih ederler [3].

Yapılan standart tanımlarına göre akıllı kartlarda bulunabilecek dosya tipleri üç başlık altında toplanabilir. Kişisel bilgisayarlardaki dosya organizasyonu ile benzeştirmek gerekirse bu sistemlerdeki dosyalara akıllı kartlar üzerindeki 'Temel Dosyalar' (elementary files) karşılık gelir. 'Atanmış Dosya' (dedicated file) olarak adlandırılan dosya tipi, dizinlere benzemektedir. Fakat bu dosyalarda dizinlerden farklı olarak veriler de bulunabilir. 'Ana Dosya' (master file) ise kişisel bilgisayarlardaki sistem dosyasına benzer işleve sahiptir.

2.4. Güvenlik

Akıllı kartların kimlik doğrulama alanında alternatif ürünlere göre daha çok rağbet görmelerinin en büyük sebebi, yüksek güvenliği düşük maliyet ile sunabilmeleridir. Akıllı kart uygulamalarında birçok seviyede, farklı güvenlik mekanizmalarına gerek duyulmaktadır. Genel olarak akıllı kart uygulamalarının güvenliği, kart ve uygulama olmak üzere iki ayrı seviyede incelenir.

Uygulama seviyesinde istenen düzeyde güvenlik elde edebilmek için öncelikle kart seviyesindeki güvenliğin sağlanması şarttır. Bu aşamada dikkat edilmesi gereken iki husus bulunmaktadır:

- Kart üzerinde tutulan veriye belirlenmiş yol dışında başka herhangi bir yoldan ulaşılmayacağı kart işletim sistemi tarafından garanti edilmelidir. Bir uygulamanın kartta çalışan diğer bir uygulamanın verilerine, onay almadan ve belirlenmiş yolları izlemeden ulaşması engellenmelidir.

- Bilgiye erişmek için gereken güvenlik kontrollerini yapmak üzere işletim sistemi içinde servisler bulunması gerekmektedir. Örneğin, belirli bir dosyaya erişmeden önce şifre sorulabilir, ya da iletilen verinin şifrelenmesi istenebilir.

ISO 7816 – 7’de güvenlik için sunulan bu mekanizmalar detaylandırılmıştır [3, 13, 14].

2.5. Akıllı Kartların Uygulama Alanları

Akıllı kart uygulamaları mağaza, telefon, ev, ofis gibi kart okuyucunun kurulabildiği her ortamda kullanılabilirler. Günümüzde akıllı kart teknolojisini temel alan uygulamalar büyük bir hızla yaygınlaşmaktadır. Akıllı kartlar telefonlardan benzin pompalarına, banka servislerinden elektronik ödeme sistemlerine kadar çok geniş bir uygulama alanına sahiptir. Aşağıda akıllı kartların sıkça kullanıldığı uygulama alanlarından bazı örnekler verilmiştir.

Ödemeli Telefonlar - Önceden Ödemeli Kartlar : Telekomünikasyon şirketlerinin kart-tabanlı ödemeli telefonlara yönelmeleriyle birlikte jetonla çalışan ödemeli telefonların önemleri ve kullanım alanları büyük oranda azalmıştır. Eski jetonlu telefonlar neredeyse tamamıyla ortadan kalkmış, yerlerini kartlı telefonlar almıştır. Her yıl tüm dünyada yaklaşık 500 milyon telefon kartı satılmaktadır. Bu sistem sayesinde telekomünikasyon şirketleri daha az bir bakım maliyetiyle karşı karşıya kalırlar. Ayrıca önceden ödeme mantığı sonucunda nakit gelir akışlarında da bir artış gözlenir. Müşteriler için bozuk para temin etme sorunu ortadan kalkacağından toplam kalite ve buna bağlı olarak müşteri memnuniyeti artar. Akıllı kart kullanan telefon sistemleri ucuz olmalarına rağmen daha güvenlidirler [15].

Mobil Haberleşme - Abone Tanıma Modülü (SIM-Subscriber Identification Module) : Haberleşme endüstrisinin en hızlı gelişen sektörü olan mobil haberleşmede, akıllı kartların sunduğu özellikler sayesinde daha güvenli ve hızlı hizmet verilebilmektedir. SIM kart herhangi bir telefona takıldığında o telefon bir çok avantajla beraber kart sahibine ait duruma gelir. Akıllı kartların kullanımıyla birlikte mobil telefonlardaki sahtekarlık oranı da büyük ölçüde azalmıştır. Bu kartlar sayesinde GSM

telefonlar, abonenin şebekeye girişini üst seviyede PIN kodu ile, daha alt seviyelerde ise bazı karmaşık şifreleme algoritmaları ile denetlemektedirler. Bu tip bir sistem usulsüz kullanım riskini neredeyse tümüyle ortadan kaldırmaktadır [6, 15, 19]:

Bankacılık ve Perakende Ticaret - Kredi, Debit, Elektronik Cüzdan Kartları : Günümüzde kredi, banka veya elektronik cüzdan kartı olarak kullanılabilen akıllı kartlar, bu manyetik kartlarla gerçekleştirilen tüm finansal bankacılık işlemlerini yerine getirebilmektedirler. PIN kontrolü gibi kimlik doğrulama mekanizmaları ile hem müşteri hem de banka için daha güvenilir bir ortam hazırlarken sahtekarlıkları da neredeyse imkansız hale getirmiş olur. Ayrıca karttaki mikroişlemcinin taklit edilmesinin zor olması nedeniyle hilekarlıkta büyük oranda azalma sağlanmaktadır [6, 15, 19].

Elektronik Ödeme : Akıllı kartlar üzerinde tutulan kredi miktarı ile alışveriş, bilet, park, çamaşırhane, kütüphane, taksi vs. için nakit para yerine elektronik para ile ödeme yapılabilir. Daha güvenli olmasının yanı sıra bu yöntem sayesinde piyasadaki nakit para dolaşımı da azaltılmış olur. Tekrar kredi doldurulabilen bu kartlar ile tek bir kartla tüm bu servisler için hizmet almak mümkündür. Küçük meblağlar üzerinde işlem yapmak mümkün olduğundan, daha büyük değerlere yuvarlamalar sonucu karşılaşılan kayıplar yaklaşık %20 oranında azalmaktadır. [6, 15, 19].

Ulaşım - Elektronik Bilet : Akıllı kartların yaygın olarak kullanıldığı elektronik bilet uygulamalarının Türkiye'deki önde gelen öneği otoban gişelerinde kullanılan 'Otomatik Geçiş Sistemi'dir. Kontaksız kartlar sayesinde gişelerdeki işlem zamanı azalmaktadır. Ayrıca veriler sayısal ortamda depolandığı için ileriye yönelik istatistiksel bir kaynak da oluşturmaktadırlar. Bir tek kart ile bir çok farklı toplu taşıma sistemine erişim mümkün olduğundan, çok operatörlü ulaşım ağlarında bile son derece verimli sonuçlar alınabilmektedir [15, 19].

Otoparklar : Elektronik bilet uygulamalarına benzer bir alan da otopark ödemeleridir. Bu sistem aynı zamanda şoförler ve belediyeler açısından da bir çok avantajı beraberlerinde getirmektedirler. Para alışverişi ortadan kalktığı için ödeme süresi kısalmaktadır. Ayrıca akıllı kartlar kullanıldığında aboneler, sağlık personeli, sık

kullananlar gibi farklı kişiler için farklı fiyatlandırma yöntemleri uygulamak mümkün olmaktadır [15, 19].

Sağlık : Doküman işleme maliyetlerinin yüksek olduğu sağlık uygulamalarında akıllı kartların kullanılmasıyla birlikte bir çok avantaj sağlanmıştır. Bilgisayarlı terminaller ve kişiselleştirilmiş hasta kimlik kartları sayesinde kağıt üzerinde tutulması gereken veriler azalmakta, bunun yanında acil durumlarda hasta ile ilgili verileri çok daha çabuk ulaşıldığı için sağlık personelinin daha hızlı ve doğru karar vermesine yardımcı olunmaktadır. Ayrıca sağlık hizmetlerinin kötü amaçlı kişiler tarafından kullanılması suretiyle yapılabilen hilekarlıklar son bulmakta ve tümüyle güvenli sistemler kurulduğundan sağlık personelinin sadece yetkisi olan bilgilere erişimi kolayca sağlanmaktadır [15, 19].



3. İTÜKART UYGULAMASI

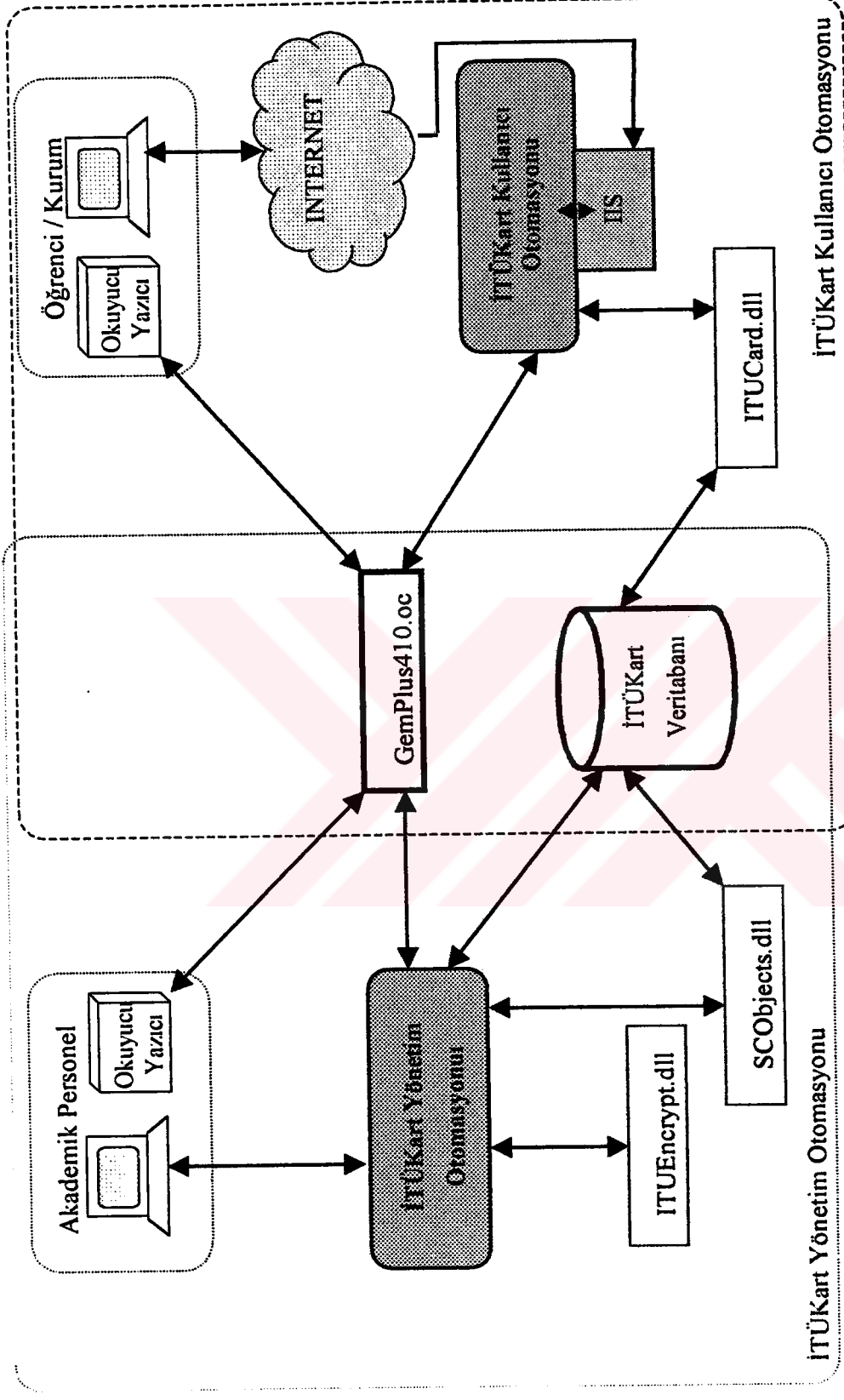
Akıllı Kartlar dünyada birçok üniversitede kimlik kartı olarak kullanılmaktadır. Tez çalışması olarak gerçekleştirilen İTÜKart projesi, benzer bir kimlik kartı uygulamasını Türkiye'ye taşımak amacıyla örnek uygulamaların temel özelliklerini barındıran bir prototip olarak tasarlanmıştır. Proje, Yönetim Otomasyonu ve Kullanıcı Otomasyonu olmak üzere iki alt bölümden oluşmaktadır. Şekil 3.1'de sistem mimarisi verilmiştir [1].

Bu tez çalışması kapsamında ele alınan İTÜKart Yönetim Otomasyonu, İTÜKart sisteminin ana veri kaynağıdır. Kampüs, bina, sınıf, bölüm, ders, kullanıcı gibi sistemin çalışması için gerekli tüm temel tanımlar bu program aracılığıyla Kullanıcı Otomasyonu ile ortak kullanılan veritabanına yazılır. Temel tanımların yanısıra genel parametreler, ders programları, sınav programları, ders başarı notları gibi sistemde sürekli güncellenen bilgilerin girişi de Yönetim Otomasyonu üzerinden yapılır.

İTÜKart Yönetim Otomasyonu ile yapılan işlemlerin önemi göz önüne alındığında bu programın yetkili kişilerce kullanılması gerektiği ortadadır. Bunun sağlanması için program exe olarak hazırlanmıştır ve otomasyon kullanıcılarının bilgisayarlarına kolaylıkla yüklenebilir. Ayrıca bu bilgisayarlarda bulunan akıllı kart okuyucu sayesinde kullanıcı doğrulaması da yapılmaktadır.

Yönetim Otomasyonu'nda yer alan kart okuyucu ile ilgili işlemler tez kapsamında yazılan ve aynı zamanda İTÜKart Kullanıcı Otomasyonu tarafından kullanılan, GemPlus410 ActiveX kontrolü üzerinden gerçekleştirilmektedir [1].

Öğrencilere yönelik ders seçme, ders durumu takibi, kullanıcı bilgileri görüntüleme, sınıf randevusu gibi özellikler ile tüm sistem kullanıcılarına yönelik şifre değiştirme, ödeme gibi olanaklar ise İTÜKart Kullanıcı Otomasyonu kapsamında ele alınmıştır[1].



Şekil 3.1 İTÜKart Genel Sistem Mimarisi

3.1. Yazılım ve Donanım Gereksinimleri

Akıllı kart tabanlı bir veritabanı uygulaması olan İTÜKart Yönetim Otomasyonu'nda sisteminin çalışması için gereken tanımlar yapıldığından program sadece yetkili kişiler tarafından kullanılmalıdır. Programın kullanılacağı bilgisayarlarda kimlik kartlarına erişimi sağlayacak GemPlus kart okuyucu mutlaka bulunmalıdır. Projenin geliştirilmesi sırasında Gemcore 410 kart okuyucu kullanılmıştır. Kimlik kartları ise yine GemPlus firmasının bir ürünü olan GemClub (1K) akıllı kartları üzerinde geliştirilmiştir. Otomasyonun kart okuyucu ile iletişimini sağlayan arabirim programı okuyucu ile birlikte gelen standart kütüphane fonksiyonlarını kullandığı için İTÜKart Yönetim Otomasyonu yüklenmeden önce kart okuyucunun söz konusu bilgisayara kurulması gerekmektedir.

Programın performanslı çalışması için uygulama modüllerinin çalışacağı bilgisayarların en az 100 MHz hıza sahip Pentium (veya dengi) işlemci ve 64 MB belleğe sahip olması gerekmektedir. Ayrıca, kart okuyucunun bilgisayara bağlanacağı kullanılmayan boş bir harici seri porta da ihtiyaç vardır. Port bilgisi arabirim programında verildiği için uygulamanın kullanılacağı tüm bilgisayarlarda aynı port kullanılmalıdır. Proje geliştirme aşamasında kart okuyucu COM1'e bağlanmıştır.

Merkezi veritabanının bulunduğu ana bilgisayar ise, 400 MHz ve üstü hızında Pentium (veya dengi) işlemciye ve en az 256 MB RAM'e sahip olmalıdır. Bu makinada MS SQL Server 7.0'ın kurulması için gereken yaklaşık 150 MB dışında bütün verileri tutabilecek büyüklükte sabit disk alanı gerekmektedir. MS SQL Server 7.0 kurmak için bilgisayarda işletim sistemi olarak Windows NT Server 4.0, Windows 98 veya Windows 2000 bulunmalıdır. İTÜKart Kullanıcı Otomasyonu kapsamında kullanılacak olan ve Windows NT Server 4.0 veya Windows 2000 üzerinde çalışan IIS yazılımının da aynı ana bilgisayara yüklenmesi durumunda SQL Server'ın bu işletim sistemlerinden birinin üzerinde çalışması daha uygundur.

3.2. Kullanılan Teknolojiler ve Diller

GemCore akıllı kart okuyucu içeren terminallerde çalışacak olan Yönetim Otomasyonu, Visual Basic 6.0 ile “Standard EXE” projesi olarak geliştirilmiştir. Visual Basic ile hazırlanan kurulum disketleri sayesinde program yetkili kişilerin PC’lerine kolayca kurulabilmektedir.

İTÜKart Kullanıcı Otomasyonu ile ortak kullanılan ve merkezi bir bilgisayarda MS SQL Server 7.0’da tutulan veritabanına erişim ADO (ActiveX Data Objects) teknolojisi ile sağlanmıştır. Yine Visual Basic 6.0 ile geliştirilen ActiveX DLL ile ADO bağlantısı üzerinden veritabanına bağlanmak, sorgulama veya değişiklik yapmak mümkün olmaktadır [1].

İTÜKart projesinde kart kullanıcılarının, biri kart üzerinde biri de veritabanında olmak üzere iki şifresi bulunmaktadır. Veritabanında tutulan şifre veri güvenliğini arttırmak amacıyla şifrelenerek saklanmaktadır. Yönetim Otomasyonu içinde kullanıcı doğrulaması yapılırken ekrandan girilen şifre ile veritabanında tutulan şifrenin karşılaştırılması sırasında uygulanan şifreleme/şifre çözme işlemleri de ayrı bir ActiveX DLL kullanılarak gerçekleştirilmiştir. Böylece şifreleme algoritmasının program kodu içinde görülmesiyle oluşacak güvenlik açıkları engellenmiştir.

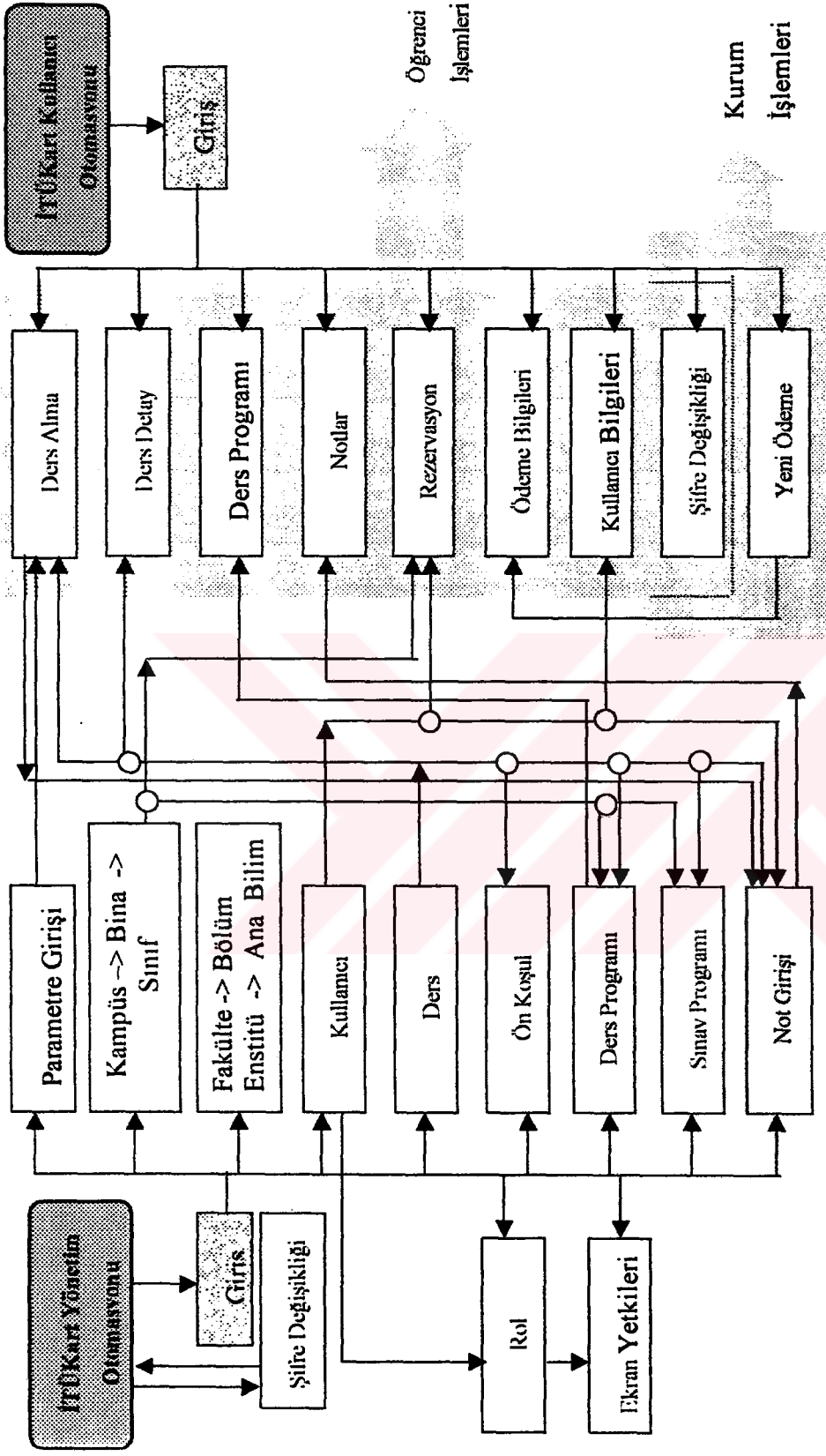
Akıllı kart okuma/yazma işlemleri için kullanılan ve terminal ile kart okuyucu arasındaki arabirim görevini yerine getiren ‘GemPlus410’ yazılımı, Visual Basic 6.0’da “ActiveX Kontrol” olarak geliştirilmiştir. Bu arayüz programında GemCore 410 kart okuyucunun kütüphane fonksiyonları kullanılmaktadır.

3.3. İşlevler

İTÜKart Yönetim Otomasyonu’nun temel işlevleri şöyle özetlenebilir:

- **Sistem Parametreleri Tanımı** : Akademik takvimde yer alan kayıt başlangıç – bitiş tarihleri, final başlangıç – bitiş tarihleri gibi tarihler tanımlanır.

- **Kampüs–Bina–Sınıf Tanımları :** İTÜKart projesinde kullanılacak sınıf, bina ve kampüs tanımları yapılır. Sınıf tanımları ders programı, sınıf programı ve randevu modüllerinde kullanılır [1].
- **Bölüm/Ana Bilim Dalı–Fakülte/Enstitü Tanımları :** İTÜKart kullanıcılarının bağlı oldukları bölüm, ana bilim dalı, fakülte ve enstitüler Yönetim Otomasyonu’nda tanımlanır. Burada yapılan bölüm ve ana bilim dalı tanımlarına daha sonraki kullanıcı ve ders tanımları sırasında başvurulur.
- **Kullanıcı Tanımı :** İTÜKart sistemine dahil tüm kullanıcılar (öğrenci, öğretim üyesi, kurum çalışanları, fakülte/enstitü çalışanları) Yönetim Otomasyonu kullanılarak tanımlanır. Buradan yapılan kullanıcı tanımları daha sonra bölüm, ana bilim dalı, fakülte, enstitü, ders, ders notları tanımlarında ve randevu işlemlerinde kullanılır [1].
- **Ders/Ön Koşul Tanımı :** Üniversitede verilen ya da verilecek dersler otomasyon yardımıyla sisteme tanımlanır. Tanımlanan dersin ön koşulu olan bir ders varsa bu bilgi de sisteme eklenir. Tanımlanan dersler daha sonra ders programı, sınav programı ve ders notu girişinde kullanılmaktadır.
- **Ders Programı Tanımı :** Üniversitede yürürlükte olan ders programları Yönetim Otomasyonu üzerinden tanımlanır. Öğrencilerin ders seçimi sırasında bu bilgiler kullanılır [1].
- **Sınav Programı Tanımı :** Yapılan tüm sınavlar (kısa sınav, vize, final) sistemde tanımlanır.
- **Ders Notu Girişi :** Yapılan sınavların sonuçları ilgili öğretim üyesi tarafından sisteme aktarılır. Böylece öğrenciler İTÜKart Kullanıcı Otomasyonu ile ders durumlarını takip edebilirler [1].



Şekil 3.2. Yazılım fonksiyonlarının genel giriş-çıkış şeması

- **Rol Tanımı :** İTÜKart Yönetim Otomasyonu kapsamında bir kullanıcı tüm ekranlara giremez; sadece yetkili olduğu ekranları açabilir. Ekran yetkileri kullanıcı rollerine verilmiştir. Bir kullanıcının belirli bir ekrana girebilmesi için o ekrana yetkisi olan bir role sahip olması gerekmektedir. Kullanıcılara rollerin atanması da sistem yöneticisi tarafından Yönetim Otomasyonu kullanılarak yapılır.
- **Ekran yetkileri :** Kullanıcı rollerine ekran girişi için yetkiler sistem yöneticisi tarafından Otomasyon üzerinden verilir.

Şekil 3.2’de otomasyon yazılımlarının fonksiyonel giriş/çıkış şeması verilmiştir.

3.4. Veritabanı

İTÜKart Yönetim Otomasyonu ve İTÜKart Kullanıcı Otomasyonu tarafından ortak olarak kullanılan merkezi veritabanı MS SQL Server 7.0 üzerinde yapılandırılmıştır.

3.4.1. Tablolar

İTÜKart projesinde Yönetim Otomasyonu ile Kullanıcı Otomasyonu aynı veritabanı üzerinde çalışmaktadır. Otomasyonlar arası veri akışı veritabanında yer alan tablolar üzerinden sağlanmaktadır. Ortak veritabanında Yönetim Otomasyonu tarafından aşağıda belirtilen tablolar kullanılmaktadır. Bu tabloların yapısı Ek.B’de verilmiştir.

- TBL_BINA
- TBL_BOLUM
- TBL_DERS
- TBL_DERSPRG
- TBL_FAKULTE
- TBL_HATALAR

- TBL_KAMPUS
- TBL_KULLANICI
- TBL_OGRDERS
- TBL_ONKOSUL
- TBL_PARAM
- TBL_ROL
- TBL_ROLEKRAN
- TBL_SABIT
- TBL_SINAVPRG
- TBL_SINIF

3.4.2. Bağımlılıklar

İTÜKart Yönetim Otomasyonu tarafından kullanılan veritabanı tabloları arasındaki bağımlılıklar Şekil 3.3'te verilmiştir.

3.5. Güvenlik

İTÜKart projesindeki verilere iki şifre üzerinden erişilmektedir. Veritabanındaki kullanıcı şifresi ile kart üzerinde yer alan PIN birlikte kullanılarak veri güvenliğini sağlamak üzere iki katmanlı bir yapı oluşturulmuştur. Kartı ele geçiren bir kişinin verilere ulaşması için kullanıcı kodu, kullanıcı şifresi ve kart PIN'ini bilmesi gerekmektedir. Kart üzerinde tutulan hatalı giriş sayısı, maksimum hatalı giriş sayısı parametresini geçerse kart kullanılamaz hale gelir. Maksimum hatalı giriş sayısı yetkili kişilerce değiştirilebilir. En fazla 15 değerini alabilen bu parametreye şu an için İTÜKart sisteminde de bu değer atanmıştır.

Kart üzerindeki bilgilere erişmek için, önce program giriş ekranından girilen kullanıcı şifresi ile veritabanında tutulan şifre karşılaştırılır. Veritabanındaki değer ile ekrandan girilen değerın şifrelenmiş hali aynı değilse programa girilemez.

Kullanıcı şifresinin başarı ile verilmesi ekranlara girmek için yeterli değildir. Kullanıcının kart şifresini (PIN) de doğru olarak vermesi gerekmektedir. Kullanıcının girdiği PIN ile kartta tutulan bilgi karşılaştırılır. Bilgiler aynı ise kullanıcının yetkili olduğu ekranlara girmesine izin verilir.

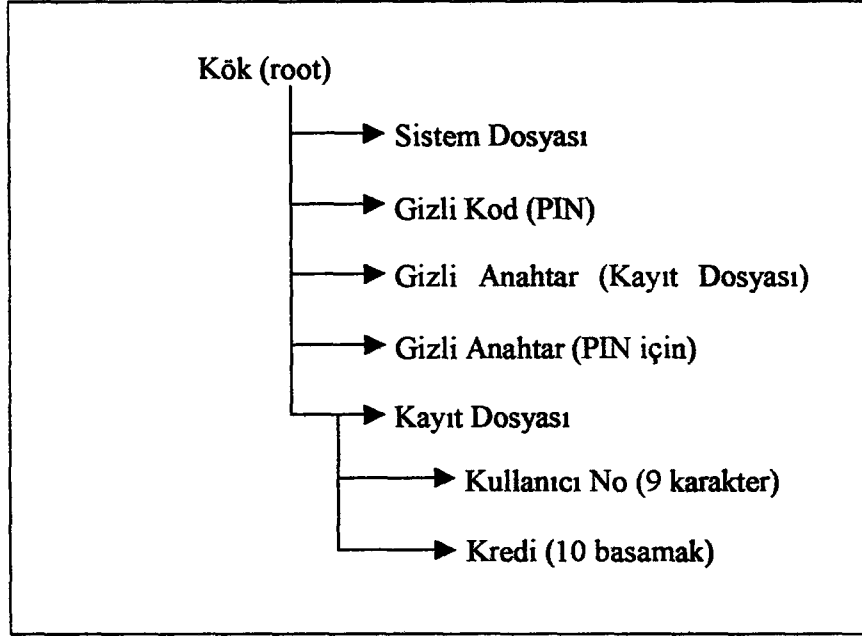
Kartta tutulan PIN ve diğer bilgiler ayrıca kartta yer alan farklı anahtarlar ile korunmaktadır. Yönetim Otomasyonu içerisinde çağırılan arabirim metodları ile PIN değişimi yapılırken PIN'i koruyan gizli anahtar değerinin de doğru olarak verilmesi gerekmektedir. Benzer şekilde kart üzerinde tutulan kayıt dosyası da koruma altındadır. Kayıt dosyasında tutulan kullanıcı kodu ve kredi bilgilerini okumak için PIN, değiştirmek için ise hem PIN hem de ilgili gizli anahtar değerinin program tarafından verilmesi gerekmektedir.

3.6. Modüller

İTÜKart Yönetim Otomasyonu kapsamında hazırlanan tüm modüller Visual Basic 6.0 kullanılarak hazırlanmıştır.

3.6.1. GemPlus410 Arayüz Kontrolü

Bilgisayar ile GemClub kart okuyucu arasındaki iletişimi sağlayacak fonksiyonlar ActiveX kontrol yapısında gerçekleştirilmiştir. Böylece aynı metodların hem Visual Basic ile hazırlanan İTÜKart Yönetim Otomasyonu'ndan hem de web ortamında hazırlanan İTÜKart Kullanıcı Otomasyonu'ndan çağırılabilmesi mümkün olmuştur.



Şekil 3.4 İTÜKart için kullanılan dosya yapısı

İTÜKart uygulamasında kullanılan GemClub kartı üzerinde kullanıcı numarası ve kullanıcının o an sahip olduğu kredi tutarı saklanmaktadır. Verilerin güvenliği için kartın kendisi tarafından sağlanan PIN mekanizmasından yararlanılmıştır. Kartta tutulan her iki verinin okunması işlemi PIN, güncellenmesi işlemi ise hem PIN hem de ilgili kayıt dosyasına ait gizli kod ile korunmaktadır. Ayrıca PIN'in değiştirilmesi için de farklı bir gizli kod kullanılmaktadır.

Hazırlanan bu genel amaçlı kontrol nesnesi, GemClub kart ve kart okuyucunun kullanıldığı başka uygulamalarda da proje referanslarına eklenerek kullanılabilir.

3.6.1.1.Özellikler

Public KartSeriNo As String : Okuyucudaki kartın seri numarası. Değeri oturum açıldığında atanır.

Public KullaniciNo As String: Karttaki kayıt dosyasında bulunan 9 karakterlik kullanıcı numarası.

Public Kredi As String : Karttaki kayıt dosyasında bulunan 10 basamaklı kredi tutarı.

Public HataKodu As Integer : Yapılan son işlemin döndürdüğü hata kodu.

Public HataTanimi As String : Yapılan son işlemin döndürdüğü hata tanımı.

3.6.1.2. Metodlar

OturumAc

Giriş Parametreleri : -

Çıkış Parametreleri : -

Önce kart okuyucu ile PC arasındaki yeni bir iletişim kanalı açılır. Eğer hata oluşursa nesnenin hata ile ilgili HataKodu ve HataTanimi özelliklerine atama yapılır. Ardından o kanal içinde yeni bir oturum açılır. Eger hata oluşursa açılan yeni kanal kapatılır ve nesnenin ilgili özelliklerine atama yapılır.

OturumKapat

Giriş Parametreleri : -

Çıkış Parametreleri : -

Önce kart okuyucu ile PC arasındaki mevcut oturum kapatılır. Eğer hata oluşursa nesnenin HataKodu ve HataTanimi özelliklerine atama yapılır. Ardından da oturumun bulunduğu kanal kapatılır. Benzer şekilde eğer hata oluşursa nesnenin ilgili özelliklerine atama yapılır.

PINKontrol

Giriş Parametreleri :

PINUzunlugu (PIN'in uzunluğu - Maksimum 8 karakter)

PINKodu (Kullanıcının kontrol edilmek üzere verdiği PIN değeri)

Çıkış Parametreleri :

PINKontrol (Kontrol işleminin sonucu geri döndürülür)

Verilen PIN değeri sistem dosyası içinde belirtilen gizli kod dosyasındaki PIN değeri ile karşılaştırılır. Sonuca ilişkin durum kodu geri döndürülür.

Oku

Giriş Parametreleri : -

Çıkış Parametreleri : -

Kayıt dosyasındaki Kullanıcı No ve Kredi bilgileri okunur, ilgili özellikler güncellenir. Eğer hata oluşmuşsa nesnenin ilgili özellikleri değiştirilir.

KrediYaz

Giriş Parametreleri :

pKredi (Kullanıcı tarafından verilen yeni kredi miktarı)

Çıkış Parametreleri : -

Kayıt dosyasındaki Kredi bilgisi, ilgili gizli anahtar değeri verilerek güncellenir. Eğer hata oluşmuşsa nesnenin ilgili özellikleri değiştirilir.

KNoYaz

Giriş Parametreleri :

pKullaniciNo (Kullanıcı tarafından verilen yeni kullanıcı numarası)

Çıkış Parametreleri : -

Kayıt dosyasındaki kullanıcı numarası bilgisi ilgili gizli anahtar değeri verilerek güncellenir. Eğer hata oluşmuşsa nesnenin ilgili özellikleri değiştirilir.

PinYaz

Giriş Parametreleri :

pNewPin (Kullanıcının girdiği yeni PIN değeri)

Çıkış Parametreleri : -

Gizli kod dosyasındaki PIN bilgisi ilgili gizli anahtar değeri verilerek güncellenir. Eğer hata oluşmuşsa nesnenin ilgili özellikleri değiştirilir.

3.6.1.3. Metotlarda Kullanılan Fonksiyonlar

UserControl_Initialize

Giriş Parametreleri : -

Çıkış Parametreleri : -

Kontrol nesnesi yaratıldığında tüm metotlar tarafından kullanılan hata mesajları dizisi doldurulur.

VB_KanalAc

Giriş Parametreleri : -

Çıkış Parametreleri : -

Haberleşmenin yapılacağı port koştulanır. Belirtilen özellikler ile yeni kanal açılır.

VB_OturumAc

Giriş Parametreleri : -

Çıkış Parametreleri : -

Haberleşmenin yapılacağı oturum açılır.

VB_KanalKapat

Giriş Parametreleri : -

Çıkış Parametreleri : -

Haberleşmenin yapıldığı kanal kapatılır.

VB_OturumKapat

Giriş Parametreleri : -

Çıkış Parametreleri : -

Haberleşmenin yapıldığı oturum kapatılır.

VB_KayitOku

Giriş Parametreleri :

DosyaNo (Okunacak kaydın bulunduğu kayıt dosyasının numarası)

KayitNo (Okunacak kaydın numarası)

KayitUzunlugu (Okunacak kaydın uzunluğu)

Çıkış Parametreleri :

Okunan (Okunan kayıttaki verinin değeri bu parametre ile geri döndürülür)

VB_KayitOku (Okuma işleminin sonucu geri döndürülür)

Belirtilen dosyadan, istenen uzunlukta kayıt okunarak geri döndürülür.

VB_KSNOku

Giriş Parametreleri :

Çıkış Parametreleri :

Okunan (Okunan kart seri numarası bu parametre içinde geri döndürülür)

VB_KSNOku (Okuma işleminin sonucu geri döndürülür)

Kart Seri Numarası (KSN) okunarak 'Okunan' adlı değişken içerisinde geri döndürülür.

VB_KayitGuncelle

Giriş Parametreleri :

DosyaNo (Güncellenecek kaydın bulunduğu kayıt dosyasının numarası)

KayitNo (Güncellenecek kaydın numarası)

KayitUzunlugu (Güncellenecek kaydın uzunluğu)

Yazilan (Güncelleme değeri)

GuncellemeSifresi (Güncelleme yapılacak kaydın bulunduğu dosyada güncelleme yapmak için gerekli olan gizli anahtar)

Çıkış Parametreleri :

VB_KayitGuncelle (Güncelleme işleminin sonucu geri döndürülür)

Belirtilen dosyadan, belirliten kaydın içeriği verilen 'Yazilan' isimli parametre ile değiştirilir.

VB_PINGuncelle

Giriş Parametreleri :

PINVerisi (Yeni PIN değeri)

PINUzunlugu (PIN'in uzunluđu – en fazla 8 karakter)

GuncellemeSifresi (Güncelleme yapılacak kaydın bulunduđu dosyada güncelleme yapmak için gerekli olan gizli anahtar)

Çıkış Parametreleri :

VB_PINGuncelle (Güncelleme işleminin sonucu geri döndürölür)

Sistem dosyası içinde belirtilen gizli kod dosyasındaki PIN değeri kullanıcıdan alınan yeni değeri ile güncellenir.

Byte_Karakter

Giriş Parametreleri :

ByteDizisi (Karaktere çevrilecek byte dizisi)

KayitUzunlugu (Parametre olarak verilen dizinin uzunluđu)

Çıkış Parametreleri :

Byte_Karakter (Byte dizisinin karakter karşılığı)

Verilen byte dizisi karakter katarına çevrilir.

Karakter_Byte

Giriş Parametreleri :

Katar (Byte dizisine çevrilecek karakter katarı)

KayitUzunlugu (Parametre olarak verilen katarın uzunluđu)

ByteDizisi (Karaktere çevrilecek byte dizisi)

Çıkış Parametreleri : -

Verilen karakter katarı byte dizisine çevrilir.

Bekle

Giriş Parametreleri :

Saniye (Kaç saniye beklenileceği bilgisi)

Çıkış Parametreleri : -

Verilen saniye boyunca işlem yapılmadan beklenilir.

VB_OturumDegistir

Giriş Parametreleri : -

Çıkış Parametreleri : -

Kartın okuyucudan çıkartılması gibi durumlarda aynı kanalda yeni bir oturum etkinleştirilir.

3.6.2. SObjects ActiveX DLL'i

İTÜKart Yönetim Otomasyonu'nda Visual Basic ile hazırlanan exe program içindeki veritabanı bağlantıları, ActiveX DLL olarak yazılan SObject yazılımında tanımlanan DB sınıfı kullanılarak gerçekleştirilmiştir.

3.6.2.1.Özellikler

Conn : Veritabanına bağlantıyı sağlayan nesnedir. ADODB.Connection objesinden türetilmiştir. Bu nesnenin tüm özelliklerini taşır.

3.6.2.2.Metodlar

OpenDB

Giriş Parametreleri : -

Çıkış Parametreleri :

OpenDB : Bağlantının başarı ile kurulup kurulmadığını belirtir.

İTÜKart Kullanıcı Otomasyonu ile ortak kullanılan MS SQL Server 7.0'daki ITU veritabanına bağlantıyı kurar [1]. Eğer mevcut bir bağlantı bulunuyorsa, önce eski bağlantı kapatılır, ardından yeni bağlantı açılır. OpenDB metodu mantıksal (boolean) değer döndürür. Geri dönüş değeri 'Doğru' (TRUE) ise bağlantı başarı ile kurulmuştur. Metod çağırıldığında 'Yanlış' (FALSE) değeri döndürülürse bağlantı kurulma aşamasında problem yaşanmış bağlantı doğru bir şekilde kurulamamıştır.

CloseDB

Giriş Parametreleri : -

Çıkış Parametreleri : -

Nesnenin temsil ettiği mevcut veritabanı bağlantısı sonlandırılır.

3.6.3. ITUEncrypt ActiveX DLL'i

İTÜKart Yönetim Otomasyonu içerisinde kullanıcının veritabanında tutulan şifresi ile işlem yapıldığı zaman bu bilginin şifrlenmesi gerekmektedir. Program açılışında kullanıcı doğrulaması yaparken ya da yeni kullanıcı tanımlarken ekrandan verilen bilginin şifrlenmesi için Visual Basic ile ActiveX DLL olarak hazırlanan ITUEncrypt yazılımında yer alan şifreleme ve şifre çözme metodları kullanılır.

3.6.3.1. Metotlar

Encode

Giriş Parametreleri :

Sifre : Şifrelenecek bilgi

KullaniciNo : Şifreleme sırasında kullanılacak kullanıcı numarası

Çıkış Parametreleri :

Encode : Şifreleme sonucu oluşan veri

Parametre olarak verilen dokuz basamaklı kullanıcı numarası kullanılarak bir anahtar değeri oluşturulur. Daha sonra verilen şifre bilgisi bu anahtar yardımı ile şifrelenir.

Decode

Giriş Parametreleri :

Sifre : Şifresi çözülecek bilgi

KullaniciNo : Şifre çözme sırasında kullanılacak kullanıcı numarası

Çıkış Parametreleri :

Decode : Şifre çözümü sonucu oluşan veri

Parametre olarak alınan kullanıcı numarası ile bir anahtar değeri oluşturulur. Şifreli veri bu değeri kullanarak çözülür.

3.6.4. Ekranlar

İTÜKart Yönetim Otomasyonu, Visual Basic 6.0 ile hazırlanmış toplam 20 ekrandan oluşmaktadır.

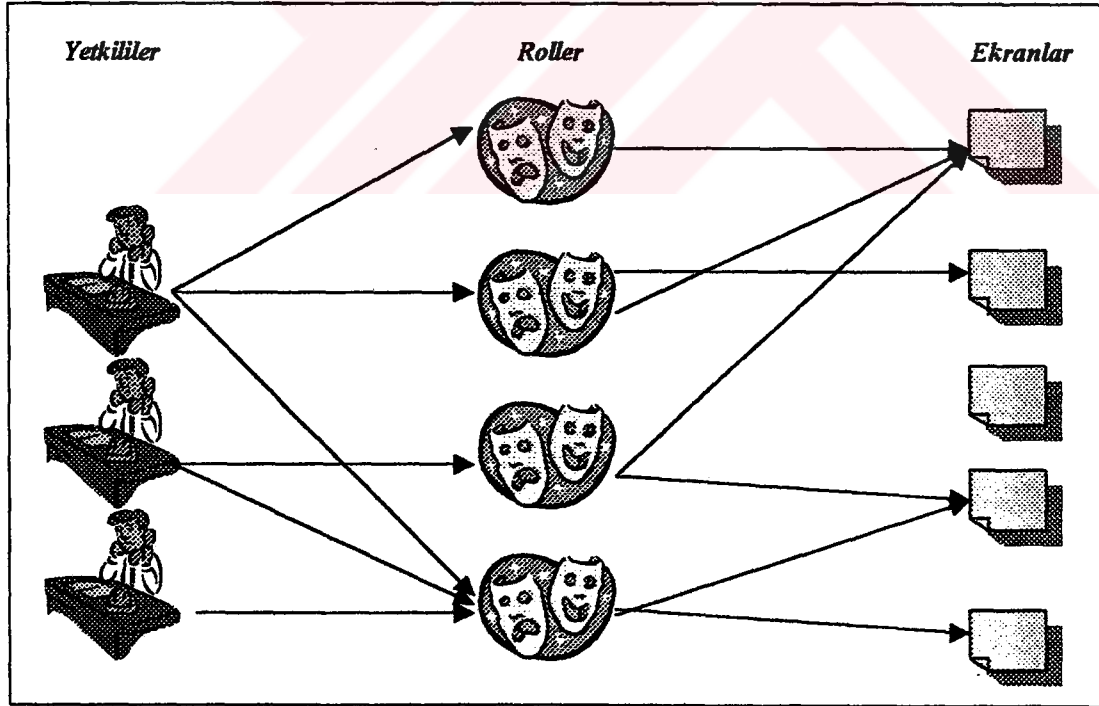
3.6.4.1.Giriş Ekranı (fLogin)

İTÜKart Yönetim Otomasyonu çalıştırıldığında ilk gelen ekrandır. Programın önemi göz önüne alındığında yetkisiz kişilerin sisteme girişlerini önlemesi gerekmektedir. Bunun için giriş ekranında 'Tamam' butonuna basıldığında, verilen kullanıcı numarasına ilişkin ünvanı kontrol edilir. Sistem yöneticisi, fakülte ve enstitü çalışanları ile öğretim görevlileri dışındaki kullanıcı numaralarının girilmesi durumunda ana ekrana geçiş yapılmaz ve hata mesajı verilir. Girişine izin verilen kullanıcılar kart PIN'i ve kullanıcı

şifreleri ile giriş yapar, sahip oldukları rollere verilmiş yetkiler ile belirlenen ekranları kullanabilirler. Şekil 3.5'te sistemdeki kullanıcı – yetki yapısı görülmektedir.

Yetkililer, Yönetim Otomasyonu'nu kullanabilen kişilerdir. Öğrenci notlarını girecek olan öğretim görevlileri ile enstitü veya fakültelerde çalışan öğrenci işleri yetkilileri İTÜKart Yönetim Otomasyonu'nun kullanıcılarıdır. Herbir yetkili aynı zamanda bir İTÜKart kullanıcısıdır, yani TBL_KULLANICI tablosunda kayıtlı bulunması gerekmektedir.

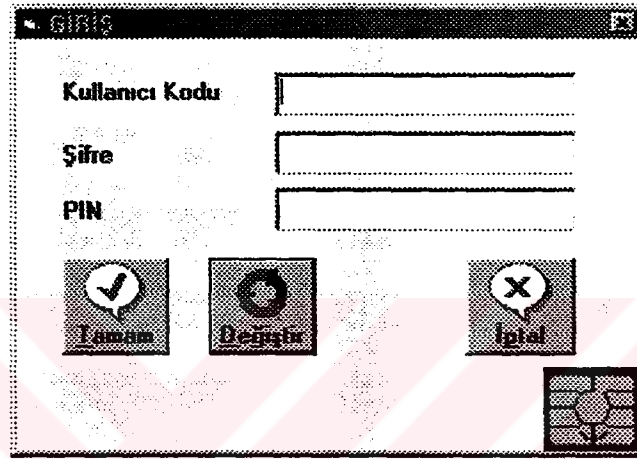
Güvenlik sebebiyle herhangi bir kullanıcı sistemdeki tüm ekranlara girememelidir. Bunu sağlamak üzere sistemde 'rol' tanımları kullanılmaktadır. Ekranlara giriş yetkileri rollere, roller ise ekranlara girilebilecek kullanıcılara verilmektedir. Örneğin, derslere ilişkin not giriş ekranına sadece öğretim üyelerinin girmesi, diğer kullanıcıların (öğrenci işleri, vs.) girmemesi isteniyorsa, sistemde bu ekrana giriş hakkına sahip bir rol yaratılıp ekranı kullanacak öğretim görevlileri bu rolle ilişkilendirilebilir. Bir role sahip kullanıcıların o rol sayesinde birden fazla ekrana giriş hakkı olabilir; bir rol birden fazla ekrana giriş yetkisi içerebilir. Ayrıca, bir kullanıcının da birden fazla rolü bulunabilir.



Şekil 3.5. Yetkili-rol-ekran yapısı

Sistemde kullanıcı tanımları TBL_KULLANICI, rol tanımları TBL_ROL, ekran yetkileri ise TBL_ROLEKRAN tablolarında tutulmaktadır.

Giriş ekranının açılışı sırasında önce SObjects dll'inde bulunan DB sınıfından türetilen bir nesne yardımıyla veritabanına bağlanılır. Ardından GemPlus410 nesnesinin hazır metodları kullanılarak GemCore kart okuyucu ile program arasında yeni bir oturum başlatılır.



Şekil 3.6. Giriş Ekranı

Şekil 3.6'da görülen Giriş ekranında kullanıcıdan kullanıcı kodu, kullanıcı şifresi ve kart PIN'i bilgisi istenir. Yazılan şifre ve PIN ekranda görüntülenmez, klavyeden tuşlara basıldıkça ilgili alana '*' karakteri basılır. Böylece kullanıcının girdiği şifrenin yetkisiz kişiler tarafından uzaktan gözlenmesi engellenir.

Kullanıcı programı kullanmaktan vazgeçmişse 'İptal' butonuna basmalıdır. 'İptal' butonuna basıldığında programdan çıkılır.

Tüm bilgiler girilip 'Tamam' butonuna basıldığında eğer kullanıcı kodu boş geçilmişse uyarı mesajı verilir. Gerekli bilgiler tam olarak verildiğinde ise belirtilen kullanıcı kodu ile TBL_KULLANICI tablosundaki şifre okunur. Kullanıcıya ait kayıt bulunamazsa "Geçersiz Kullanıcı Kodu!" hata mesajı verilir. Girilen şifre ITUEncrypt dll'indeki metodlar kullanılarak şifrelenir. Elde edilen sonuç veritabanından okunan değerle

karşılaştırılır. Eğer ekrandan girilen değer ile tablodan okunan şifrelenmiş değer farklı ise hata mesajı verilir.

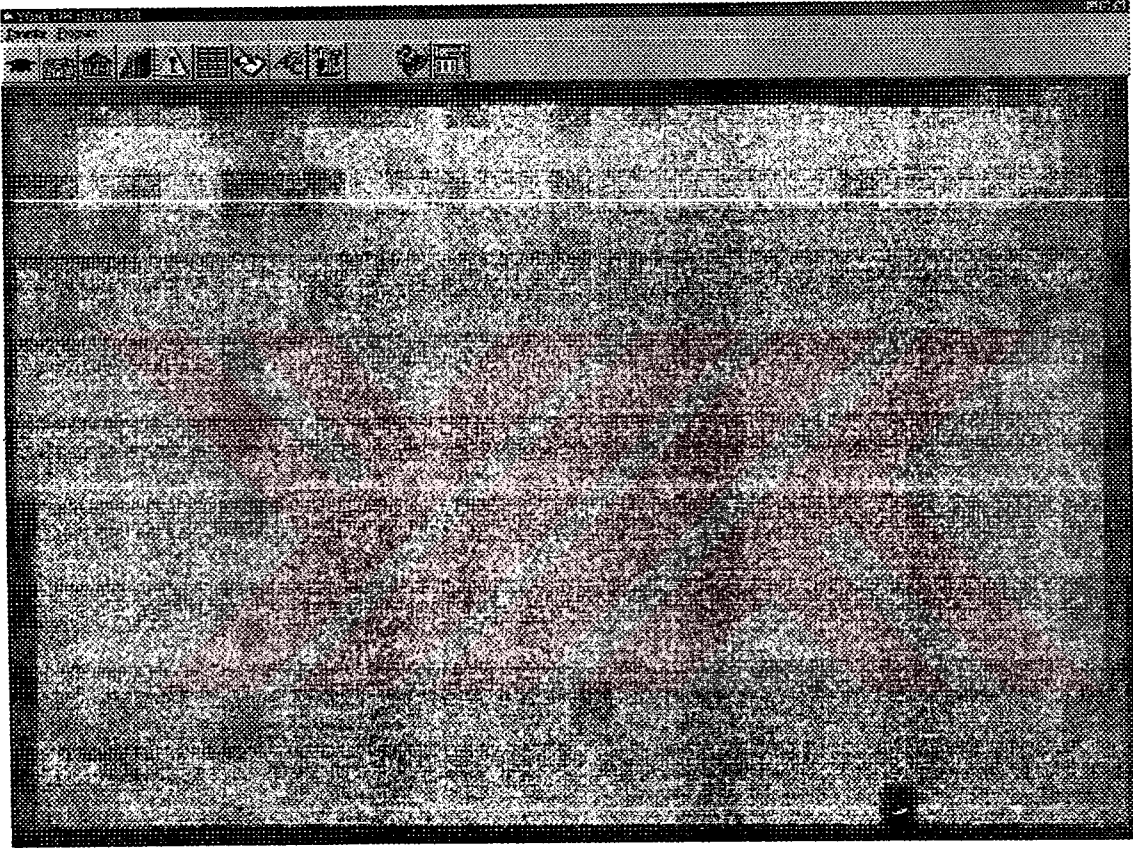
Şifrenin doğru olarak girilmesi durumunda ekranda verilen PIN, GemPlus410 arayüz nesnesinin PIN doğrulama metodu kullanılarak kontrol edilir. Verilen PIN doğru ise ana ekrana geçilir.

Şifre ve PIN değiştirmek için 'Değiştir' butonu kullanılır. Kullanıcı mevcut şifre ve PIN'ini girdikten sonra 'Değiştir' butonuna bastığında önce verdiği bilgilerin doğruluğu kontrol edilir. Bilgilerden herhangi biri yanlış ise işleme devam edilmez. Doğru şifre ve PIN girilmesi durumunda yeni bir ekran açılarak kullanıcıdan yeni şifreyi girmesi istenir. Bu ekran boş geçilirse şifre değiştirilmez. Yeni bir şifre verilirse ikinci bir ekranda bu bilginin yeniden girilmesi istenir. Girilen iki şifre farklıysa işleme devam edilmez. Yeni şifrenin düzgün bir şekilde girilmesinin ardından bu bilgi ITUEncrypt dll'indeki şifreleme metodu kullanılarak şifrelenir ve veritabanına yazılır. Şifre değişikliğinin ardından kullanıcıdan yeni PIN değeri istenir. Kullanıcı bu bilgiyi boş geçerse PIN değişikliği yapılmaz. Yeni bir PIN belirtilmesi durumunda kullanıcıdan bu bilgiyi tekrar girmesi istenir. Yeni girilen iki bilgi farklı ise işleme devam edilmez. Bilgilerin aynı olması durumunda GemPlus410 nesnesinin ilgili metodu çağırılarak değiştirme işlemi tamamlanır. Şifre veya PIN değişikliğinden sonra kullanıcı yeni değerleri ekrana yeniden girmelidir. Aksi takdirde ana ekrana geçilemez.

Sistem yöneticisinin kullanıcı kodu 999999999 olarak belirlenmiştir. Yöneticinin şifresi sabitler tablosunda şifrelenmiş halde tutulmaktadır. Sistem yöneticisinin kimlik doğrulaması veya şifre değiştirmesi sırasında sabit tablosundaki ilgili alana gidilir.

3.6.4.2.Ana Ekran (MDIForm)

Yönetim Otomasyonu'nun ana ekranı Visual Basic'te MDI Form olarak tasarlanmıştır. Diğer ekranlar, bu ana form üzerine açılan çocuk formlardır. Çocuk formlara geçişler ana formdaki toolbar üzerinde yer alan butonlar veya menü vasıtasıyla yapılır. Ana ekranın görüntüsü Şekil 3.7'de verilmiştir.



Şekil 3.7. Ana Ekran

Toolbar'daki butonlar tıklanarak sırasıyla şu ekranlara (VB Form) geçilir:

- Kullanıcı Tanımları (fOgrGrv)
- Kampüs / Bina / Sınıf Tanımları (fKBS)
- Fakülte / Enstitü / Bölüm / Ana Bilim Dalı Tanımları (fFB)

- Ders Tanımları (fDers)
- Ön Koşul Dersi Tanımları (fOn)
- Ders Programı (fDersPrg)
- Not Girişi (fNot)
- Parametre Girişi (fParam)
- Rol Tanımları (fRol)
- Ekran Yetkileri (fEkran)

Ana ekranın kapatılması ile birlikte veritabanına olan bağlantı ve kart okuyucu ile program arasındaki oturum da sonlandırılır.



3.6.4.3.Kullanıcı Tanımları (fOgrGrv, fKimlik2)

İTÜ bünyesinde hizmet veren ve İTÜKart'a sahip tüm kullanıcıların veritabanına kaydedildiği, bilgilerinin değiştirildiği veya silindiği ekrandır.

Ana Ekran (fOgrGrv)

Bu ekranda sistem kullanıcılarının kimlik bilgisi dışındaki, daha çok sistemi ilgilendiren bilgileri tanımlanır. Ekran görüntüsü Şekil 3.8'de verilmiştir.

Açıklamalar

- Ünvan alanı “”, “PROF”, “DOÇ DR”, “YRD DOÇ DR”, “DR”, “AR GOR”, “Y.LİSANS”, “LİSANS” ve “DİĞER” seçeneklerine sahiptir.
- Kullanıcı No mutlaka 9 karakter uzunluğunda, şifre ise 8 karakter uzunluğunda olmalıdır.

Fonksiyon ve Prosedürler

bCik_Click()

- Ekran kapatılır

bGetir_Click()

- Kullanıcı numarası girilmemişse hata verilip prosedürden çıkılır.
- Girilen kullanıcı numarası ile EkranDoldur prosedürü çağırılır.

Kullanıcı Tanımları

Kullanıcı No			
Ad - Soyad			
Unvan			
Giriş Yılı			
Ana Bilim Dalı			
Bölüm			
Adres			
Telefon			
Email			
Resim			
İmza			
Cinsiyet	☐ Kız ☐ Erkek		
Uyruk	☐ T.C. ☐ Diğer		
Dahili No			
Oda No			
Şifre			

Şekil 3.8. Kullanıcı Tanımları – Ana Ekran

bOku_Click()

- Okuyucuya yerleştirilen İTÜKart'tan, GemPlus410.ocx kontrolünden türetilen ekrandaki SCC1 nesnesinin 'Oku' metodu çağırılarak karttaki kullanıcı numarası okunur.
- Alınan numara ile EkranDoldur çağırılır.

bKaydet_Click()

- Kullanıcı numarası, adı, soyadı ve giriş yılı bilgileri kontrol edilir. Boş geçilen alan varsa hata verilip prosedür sonlandırılır.
- Uyruk alanında “Diğer” seçilmişse ama ilgili textbox’a giriş yapılmamışsa hata verilir ve prosedürden çıkılır.
- Şifre alanının uzunluğu sekizden farklı ise hata verilir ve prosedürden çıkılır.
- TBL_KULLANICI tablosu verilen kullanıcı numarası için okunur.
- Kayıt bulunamazsa, INSERT cümlesiyle kayıt yaratılır. Ekranda boş geçilen alanlara NULL değer atanır. Kimlik bilgilerinin girilmesi için uyarı mesajı verilir.
- Kayıt varsa, UPDATE cümlesiyle kayıt ekrandaki bilgiler ile güncellenir. Ekranda boş geçilen alanlara NULL değer atanır.

bKimlik_Click()

- Kimlik Bilgileri Ekranı yüklenir.

bSil_Click()

- Kullanıcı numarası girilmemişse hata verilir, prosedürden çıkılır.
- Girilen kullanıcı numarası ile TBL_KULLANICI okunur. Kayıt bulunamazsa hata verilerek işlem sonlandırılır.
- Kullanıcıya kaydı silmek isteyip istemediği sorulur. “Hayır” vevabı gelirse silme işlemi yapılmaz.
- “Evet” yanıtı gelirse DELETE cümlesi ile TBL_KULLANICI tablosundaki ilgili kayıt silinir. Kaydın silindiği kullanıcıya bildirilir. EkranSil prosedürü çağrılır.

bYeni_Click()

- EkranSil prosedürü çağrılır.

cmb_Bolum_Click()

- Bölüm kodu alanı boş değilse,
 - TBL_BOLUM tablosundan seçilen bölümün adı okunur.
 - Kayıt bulunursa, AD alanı ilgili textbox'a yazılır.
 - Bulunamazsa, ekrandaki Ad alanı silinir.
- Bölüm kodu alanı boş ise,
 - Bölüm adı alanı temizlenir.

cmb_ABDali_Click()

- Ana bilim dalı kodu alanı boş değilse,
 - TBL_BOLUM tablosundan seçilen ana bilim dalının adı okunur.
 - Kayıt bulunursa, AD alanı ilgili textbox'a yazılır.
 - Bulunamazsa, ekrandaki Ad alanı silinir.
- Ana bilim dalı kodu alanı boş ise,
 - Ana bilim dalı adı alanı temizlenir.

Form_Load()

- TBL_BOLUM tablosu BOLUM = 0 için okunur. Elde edilen kayıtlar ana bilim dalı combobox'ına yazılır. Combobox'da kayıt varsa ilki seçilir. (cmb_ABDali_Click çalışır.)

- TBL_BOLUM tablosu BOLUM = 1 için okunur. Elde edilen kayıtlar bölüm combobox'ına yazılır. Combobox'da kayıt varsa ilki seçilir. (cmb_Bolum_Click çalışır.)
- TBL_SABIT tablosu ALAN = 'UNVANNO' için okunur. Elde edilen kayıtlar ünvan combobox'ına yazılır. Combobox'da kayıt varsa ilki seçilir. (cmb_Unvan_Click çalışır.)

EkranDoldur(SicilNo As String)

- EkranSil prosedürü çağırılır.
- Parametre ile verilen kullanıcı numarası ile TBL_KULLANICI tablosu okunur.
- Kayıt bulunamazsa hata verilerek prosedür sonlandırılır.
- Kayıt varsa, okunan değerler ekrandaki ilgili alanlara yazılır. Kimlik bilgileri değişkenlere alınır. Tablodan NULL olarak okunan değerler alanlara ya da değişkenlere "" (boşluk) olarak aktarılır.

EkranSil()

- Ekrandaki textbox alanlar silinir.
- Combobox'ların varsa ilk kayıtları seçilir. (cmb_Bolum_Click, cmb_ABDali_Click çalışır.)
- Cinsiyet olarak "Kız", uyruk olarak "TC" radiobutton'ları seçilir.
- Kimlik bilgisi değişkenleri temizlenir.

Kimlik Bilgileri Ekranı (fKimlik2)

Kullanıcının demografik özelliklerinin görüntüldüğü, değiştirildiği bu ekranın görüntüsü Şekil 3.9’da verilmiştir.

Kimlik Bilgileri

Sicil No

Ad / Soyad

Nüfus Seri No

Baba Adı

Doğum Yeri

Medeni Hali

Ana Adı

Doğum Tarihi

Dini

Kayıt Olduğu

İl

Mahalle / Köy

Aile Sıra No

İlçe

Cilt No

Sıra No

Tamam

İptal

Şekil 3.9. Kimlik Bilgileri Ekranı

Açıklamalar

- Medeni hal alanı "", "BEKAR", "EVLİ", "DUL" seçeneklerine sahiptir.

Fonksiyon ve Prosedürler

bİptal_Click()

- Ekran kapatılır

bKaydet_Click()

- Ekrandaki değerler fOgrGrv formunun ilgili değişkenlerine atanır.
- Ekran kapatılır

Form_Load()

- fOgrGrv formundaki ad alanı dolu ise,
 - fOgrGrv formunun kimlik bilgisi değişkenlerinin içeriği ekrandaki ilgili alanlara atanır.



3.6.4.4. Kampüs, Bina, Sınıf Tanımları (fKBS)

İTÜ içinde yer alan kampüs, bina ve sınıfların tanımlarının yaratıldığı, güncellendiği ve silindiği ekrandır.

Fonksiyon ve Prosedürler

bCik_Click()

- Ekran kapatılır

Form_Load()

- Kampüs tab'ı seçilir.

TabStrip1_Click()

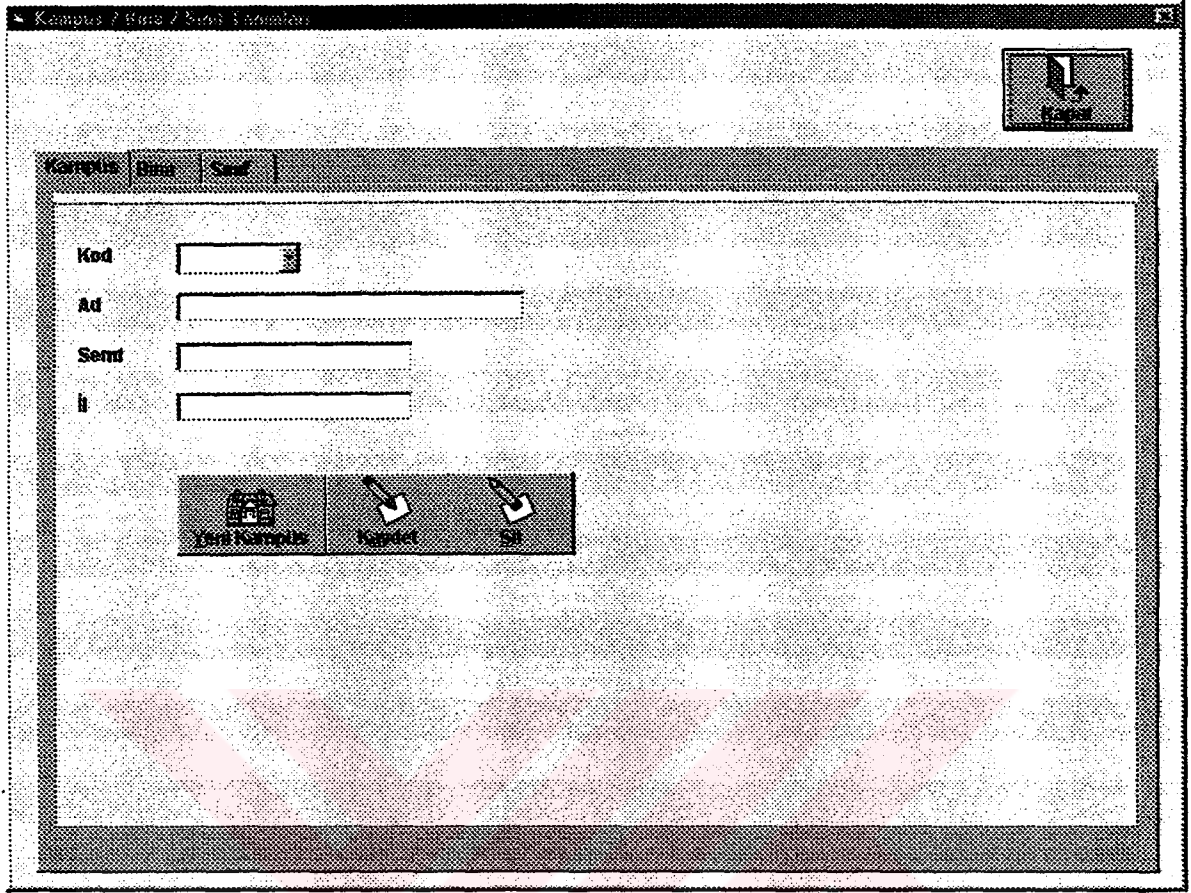
- Seçilen tab'a ait frame ekrana getirilir. KampusLoad, BinaLoad, SinifLoad prosedürlerinden ilgili olanı çağrılır.

Kampüs Tab'ı Fonksiyon ve Prosedürleri

Kampüs tanımlarının yapıldığı, değiştirildiği ve silindiği bölüm Şekil 3.10'da görülmektedir.

KampusLoad()

- TBL_KAMPUS tablosundaki tüm kayıtlar okunur.
- Kayıt bulunursa, ilk kayda ait ad, semt, il bilgileri ekrana yazılır.
- Okunan tüm kampüs kodları kod combobox'ına yazılır.
- Combobox'da kayıt varsa ilki seçilir. (cmb_Kkod_Click çalışır.)



Şekil 3.10. Kampüs Tanımları

KampusDoldur(pKKod As String)

- Verilen kampüs kodu için TBL_KAMPUS tablosu okunur.
- Kayıt varsa, ad, semt, il alanları doldurulur.
- Kayıt bulunamazsa, hata mesajı verilir. İlgili alanlar silinerek prosedür sonlandırılır.

cmb_KKod_Click()

- Seçilen kod boşluk değilse bu değer ile KampusDoldur prosedürü çağırılır.

bKYeni_Click()

- Kampüs kodu combobox'ı ve diğer alanlar temizlenir.

bKKaydet_Click()

- Kampüs kodu alanı kontrol edilir. Bu alanın boş olmasına izin verilmez.
- Verilen kampüs kodu ile TBL_KAMPUS tablosu okunur.
- Kayıt varsa, ekrandaki bilgiler ile UPDATE cümlesi oluşturulur ve kayıt güncellenir. Ekranda boş olan alanlar NULL olarak cümleye aktarılır. Kullanıcıya, işlemin bittiği bildirilir.
- Kayıt yoksa, verilen bilgilerle INSERT cümlesi oluşturulup yeni bir kayıt yaratılır. Ekranda boş olan alanlar NULL olarak cümleye aktarılır. Kullanıcıya, işlemin bittiği bildirilir. KampusLoad prosedürü çağırılarak kod combobox'ı yeniden doldurulur. Yeni kaydın kodu ile KampusDoldur çağırılır.

bKSil_Click()

- Kampüs kodu alanı kontrol edilir. Bu alanın boş olmasına izin verilmez.
- Seçilen kampüs kodu ile TBL_BINA tablosu okunur. Kampüse kayıtlı bina varsa kampüs tanımının silinmesine izin verilmez.
- Verilen kampüs kodu ile TBL_KAMPUS tablosu okunur.
- Kayıt varsa, kullanıcıya kaydı silme isteği teyid ettirilir. “Hayır” yanıtı gelirse silme işlemine devam edilmez. “Evet” yanıtı ile DELETE cümlesi oluşturularak, kayıt tablodan silinir. Kullanıcıya işlemin bittiği bildirilir. KampusLoad prosedürü çağırılır.
- Kayıt bulunamazsa, kullanıcıya hata mesajı verilerek prosedürden çıkılır.

Bina Tab'ı Fonksiyon ve Prosedürleri

Şekil 3.11'de görülen Bina bölümünde kampüslerde bulunan binalar tanımlanır.

Kampüs / Bina / Bina Tanımları

Kod

Ad

Kampüs

Yeni Bina Kaydet Sil

Şekil 3.11. Bina Tanımları

BinaLoad()

- TBL_KAMPUS tablosu okunur. Okunan tüm kodlar kampüs combobox'ına yazılır. Combobox'da kayıt varsa ilki seçilir.
- TBL_BINA tablosundaki tüm kayıtlar okunur. Kayıt yoksa, bina adı ve kampüs kodu alanları temizlenir.
- Kayıt bulunursa, okunan tüm kayıtlar bina kodu combobox'ına yazılır. Combobox'da kayıt varsa ilki seçilir. (cmb_Bkod_Click çalışır.)

BinaDoldur(pBKod As String)

- Verilen bina kodu için TBL_BINA tablosu okunur.
- Kayıt varsa, ad ve kampüs alanları doldurulur.
- Kayıt bulunamazsa, hata mesajı verilir. İlgili alanlar silinerek prosedür sonlandırılır.

cmb_BKod_Click()

- Seçilen kod boşluk değilse bu değer ile BinaDoldur prosedürü çağırılır.

bBYeni_Click()

- Bina kodu combobox'ı ve ad alanı temizlenir.
- Varsa ilk kampüs seçilir.

bBKaydet_Click()

- Bina kodu alanı kontrol edilir. Bu alanın boş olmasına izin verilmez.
- Verilen bina kodu ile TBL_BINA tablosu okunur.
- Kayıt varsa, ekrandaki bilgiler ile UPDATE cümlesi oluşturulur ve kayıt güncellenir. Ekranda boş olan alanlar NULL olarak cümleye aktarılır. Kullanıcıya, işlemin bittiği bildirilir.
- Kayıt yoksa, verilen bilgilerle INSERT cümlesi oluşturulup yeni bir kayıt yaratılır. Ekranda boş olan alanlar NULL olarak cümleye aktarılır. Kullanıcıya, işlemin bittiği bildirilir. BinaLoad prosedürü çağırılarak kod combobox'ı yeniden doldurulur. Yeni kaydın kodu ile BinaDoldur çağırılır.

bBSil_Click()

- Bina kodu alanı kontrol edilir. Bu alanın boş olmasına izin verilmez.

- Seçilen bina kodu ile TBL_SINIF tablosu okunur. Binaya kayıtlı sınıf varsa bina tanımının silinmesine izin verilmez.
- Verilen bina kodu ile TBL_BINA tablosu okunur.
- Kayıt varsa, kullanıcıya kaydı silme isteği teyid ettirilir. “Hayır” yanıtı gelirse silme işlemine devam edilmez. “Evet” yanıtı ile DELETE cümlesi oluşturularak, kayıt tablodan silinir. Kullanıcıya işlemin bittiği bildirilir. BinaLoad prosedürü çağırılır.
- Kayıt bulunamazsa, kullanıcıya hata mesajı verilerek prosedürden çıkılır.

Sınıf Tab’ı Fonksiyon ve Prosedürleri

Şekil 3.12’de sınıf tanımlarıyla ilgili işlemlerin yapıldığı Sınıf bölümü gösterilmiştir.

SınıfLoad()

- TBL_BINA tablosu okunur. Okunan tüm kodlar bina combobox’ına yazılır. Combobox’da kayıt varsa ilki seçilir. (cmb_Sbina_Click çalışır.)

SınıfDoldur(pSKod As String)

- Verilen sınıf ve bina kodları için TBL_SINIF tablosu okunur.
- Kayıt varsa, ad, kapasite ve tip (oda/sınıf/laboratuar) alanları doldurulur.
- Kayıt bulunamazsa, hata mesajı verilir. İlgili alanlar silinerek prosedür sonlandırılır.

cmb_SBina_Click()

- Seçilen bina adı varsa,
 - Binaya ait sınıflar TBL_SINIF tablosundan okunur.
 - Okunan sınıf kodları kod combobox’ına yazılır.
 - Kayıt varsa, ilki seçilir.

- Kayıt yoksa ilgili alanlar temizlenir.

Şekil 3.12. Sınıf Tanımları

cmb_SKod_Click()

- Seçilen sınıf ve bina kodları boşluk değilse bu değerler ile SinifDoldur prosedürü çağırılır.

bSYeni_Click()

- Sınıf kodu combobox'ı ve ad alanı temizlenir.
- Varsa ilk bina kaydı seçilir.

bSKaydet_Click()

- Sınıf kodu, bina kodu ve kapasite alanları kontrol edilir. Bu alanların boş olmasına izin verilmez.
- Verilen sınıf kodu ve bina kodu ile TBL_SINIF tablosu okunur.
- Kayıt varsa, ekrandaki bilgiler ile UPDATE cümlesi oluşturulur ve kayıt güncellenir. Ekranda boş olan alanlar NULL olarak cümleye aktarılır. Kullanıcıya, işlemin bittiği bildirilir.
- Kayıt yoksa, verilen bilgilerle INSERT cümlesi oluşturulup yeni bir kayıt yaratılır. Ekranda boş olan alanlar NULL olarak cümleye aktarılır. Kullanıcıya, işlemin bittiği bildirilir. SinifLoad prosedürü çağırılarak kod combobox'ı yeniden doldurulur. Yeni kaydın kodu ile SinifDoldur çağırılır.

bSSil_Click()

- Sınıf ve bina kodu alanları kontrol edilir. Bu alanların boş olmasına izin verilmez.
- Seçilen sınıf ve bina kodları ile TBL_SINIF tablosu okunur.
- Kayıt varsa, kullanıcıya kaydı silme isteği teyid ettirilir. “Hayır” yanıtı gelirse silme işlemine devam edilmez. “Evet” yanıtı ile DELETE cümlesi oluşturularak, kayıt tablodan silinir. Kullanıcıya işlemin bittiği bildirilir. SinifLoad prosedürü çağırılır.
- Kayıt bulunamazsa, kullanıcıya hata mesajı verilerek prosedürden çıkılır.

3.6.4.5.Fakülte, Enstitü, Bölüm, Ana Bilim Dalı Tanımları (fFB)

İTÜ bünyesindeki Fakülte, Enstitü, Bölüm, Ana Bilim Dalı tanımlarının yaratıldığı, güncellendiği ve silindiği ekrandır.

Fonksiyon ve Prosedürler

bCik_Click()

- Ekran kapatılır

Form_Load()

- Fakülte tab'ı seçilir.

TabStrip1_Click()

- Seçilen tab'a ait frame ekrana getirilir. FakulteLoad, EnstituLoad, BolumLoad, ABDLoad prosedürlerinden ilgili olanı çağırılır.

Fakülte Tab'ı Fonksiyon ve Prosedürler

Fakülte tanımlarının yapıldığı bölüm Şekil 3.13'te görülmektedir.

FakulteLoad()

- TBL_BINA tablosundaki tüm kayıtlar okunur.
- Kayıt bulunamazsa, bina combobox'ı temizlenir.
- Okunan tüm bina kodları ilgili combobox'a yazılır.
- Combobox'da kayıt varsa ilki seçilir.
- TBL_FAKULTE tablosundaki FAKULTE = 1 kayıtları okunur.
- Kayıt bulunamazsa, ilgili alanlar temizlenir.

- Okunan tüm fakülte kodları ilgili combobox'a yazılır.
- Combobox'da kayıt varsa ilki seçilir.

Şekil 3.13. Fakülte Tanımları

FakulteDoldur(pFKod As String)

- Kodu verilen fakülte kaydı FAKULTE = 1 ile TBL_FAKULTE tablosundan okunur.
- Kayıt bulunursa ad, bina alanları doldurulur. Fakültede görev alan tüm profesörler dekan combobox'ına yazılır. Kayıttaki dekan combobox'da seçilir.
- Kayıt bulunamazsa, hata mesajı verilir. İlgili alanlar temizlenir.

cmb_FKod_Click()

- Seçilen fakülte kodu boşluk değilse bu kod ile FakulteDoldur prosedürü çağırılır.

bFYeni_Click()

- Fakülte combobox'ı ve diğer alanlar temizlenir.

bFKaydet_Click()

- Fakülte kodu alanı kontrol edilir. Eğer boş ise hata verilerek prosedürden çıkılır.
- Verilen kod ile TBL_FAKULTE tablosu okunur.
- Kayıt varsa,
 - Ekrandaki bilgiler ile UPDATE cümlesi oluşturulur. Boş olan alanlar NULL olarak alınır.
 - Kayıt güncellenir.
 - Kullanıcıya işlemin bittiğini belirten mesaj verilir.
- Kayıt yoksa,
 - Ekrandaki bilgiler ile INSERT cümlesi oluşturulur. Boş olan alanlar NULL olarak alınır. FAKULTE = 1 olarak alınır.
 - Kayıt güncellenir.
 - Kullanıcıya işlemin bittiğini belirten mesaj verilir.
 - FakulteLoad prosedürü çağırılarak kod combobox'ının güncellenmesi sağlanır.
 - Yeni eklenen fakülte kodu ile FakulteDoldur prosedürü çağırılarak yeni kayıt ekrana getirilir.

bFSil_Click()

- Fakülte kodu alanı kontrol edilir. Eğer boş ise hata verilerek prosedürden çıkılır.
- Verilen kod ile TBL_BOLUM tablosu okunur. Fakülteye tanımlı bölüm varsa fakülte kaydının silinmesine izin verilmez.
- Verilen kod ile TBL_FAKULTE tablosu okunur.
- Kayıt varsa,
 - Kullanıcının silme isteğini teyid etmesi gerekmektedir.
 - Kullanıcı “Evet” derse, ekrandaki bilgiler ile DELETE cümlesi oluşturulur. Kayıt silinir. Kullanıcıya işlemin bittiğini belirten mesaj verilir.
 - FakulteLoad prosedürü çağırılır.
- Kayıt yoksa,
 - Kullanıcıya uyarı mesajı verilir.

Enstitü Tab’ı Fonksiyon ve Prosedürleri

Şekil 3.14’te Enstitü tanımlarının yapıldığı bölüm görülmektedir.

EnstituLoad()

- TBL_BINA tablosundaki tüm kayıtlar okunur.
- Kayıt bulunamazsa, bina combobox’ı temizlenir.
- Okunan tüm bina kodları ilgili combobox’a yazılır.
- Combobox’da kayıt varsa ilki seçilir.
- TBL_FAKULTE tablosundaki FAKULTE = 0 kayıtları okunur.
- Kayıt bulunamazsa, ilgili alanlar temizlenir.

- Okunan tüm enstitü kodları ilgili combobox'a yazılır.
- Combobox'da kayıt varsa ilki seçilir.

Şekil 3.14. Enstitü Tanımları

EnstituDoldur(pEKod As String)

- Kodu verilen enstitü kaydı FAKULTE = 0 ile TBL_FAKULTE tablosundan okunur.
- Kayıt bulunursa ad, bina alanları doldurulur.
- Enstitüde görev yapan tüm profesörler dekan combobox'ına yazılır. Enstitü kaydında belirtilen mevcut dekan combobox'da seçilir.

- Kayıt bulunamazsa, hata mesajı verilir. İlgili alanlar temizlenir.

cmb_EKod_Click()

- Seçilen enstitü kodu boşluk değilse bu kod ile EnstituDoldur prosedürü çağınır.

bEYeni_Click()

- Enstitü combobox'ı ve diğer alanlar temizlenir.

bEKaydet_Click()

- Enstitü kodu alanı kontrol edilir. Eğer boş ise hata verilerek prosedürden çıkılır.
- Verilen kod ile TBL_FAKULTE tablosu okunur.
- Kayıt varsa,
 - Ekrandaki bilgiler ile UPDATE cümlesi oluşturulur. Boş olan alanlar NULL olarak alınır.
 - Kayıt güncellenir.
 - Kullanıcıya işlemin bittiğini belirten mesaj verilir.
- Kayıt yoksa,
 - Ekrandaki bilgiler ile INSERT cümlesi oluşturulur. Boş olan alanlar NULL olarak alınır. FAKULTE = 0 olarak alınır.
 - Kayıt güncellenir.
 - Kullanıcıya işlemin bittiğini belirten mesaj verilir.
 - EnstituLoad prosedürü çağırılarak kod combobox'ının güncellenmesi sağlanır.
 - Yeni eklenen enstitü kodu ile EnstituDoldur prosedürü çağırılarak yeni kayıt ekrana getirilir.

bESil_Click()

- Enstitü kodu alanı kontrol edilir. Eğer boş ise hata verilerek prosedürden çıkılır.
- Verilen kod ile TBL_BOLUM tablosu okunur. Enstitüye tanımlı ana bilim dalı varsa enstitü kaydının silinmesine izin verilmez.
- Verilen kod ile TBL_FAKULTE tablosu okunur.
- Kayıt varsa,
 - Kullanıcının silme isteğini teyid etmesi gerekmektedir.
 - Kullanıcı “Evet” derse, ekrandaki bilgiler ile DELETE cümlesi oluşturulur. Kayıt silinir. Kullanıcıya işlemin bittiğini belirten mesaj verilir.
 - EnstituLoad prosedürü çağırılır.
- Kayıt yoksa,
 - Kullanıcıya uyarı mesajı verilir.

Bölüm Tab’I Fonksiyon ve Prosedürler

Sistemde bulunan bölümlerin tanımlanması Şekil 3.15’deki bölümden yapılmaktadır.

EnstituLoad()

- TBL_FAKULTE tablosundan tüm fakülteler FAKULTE = 1 koşulu verilerek okunur.
- Kayıt yoksa tüm alanlar temizlenerek prosedür sonlandırılır.
- Okunan kayıtlar fakülte combobox’ına yazılır.
- Kayıt varsa ilki seçilir. (cmb_BFakulte_Click çalışır.)

BolumDoldur(pBKod As String)

- Kodu verilen bölüm için BOLUM = 1 olan kayıt TBL_BOLUM tablosundan okunur.
- Kayıt varsa ad alanı doldurulur.
- Bölümde görev yapan profesörler başkan combobox'ına yazılır. Tablodan okunan mevcut başkan, bu combobox'dan seçilir.
- Kayıt yoksa, hata mesajı verilir. Alanlar temizlenir.

The screenshot shows a software window titled "Fakülte - Enstitü - Bölüm - Anabölümde Tanımlan". Inside the window, there are three dropdown menus labeled "Fakülte", "Bölüm", and "Başkan". Below these menus are three buttons: "Yeni Bölüm" (New Department), "Kaydet" (Save), and "Sil" (Delete). A "Kapat" (Close) button is located in the top right corner of the window.

Şekil 3.15. Bölüm Tanımları

cmb_BFakulte_Click()

- Seçilen fakülte kodu boş değilse,
 - TBL_FAKULTE tablosu fakülte kodu ve FAKULTE = 1 için okunarak AD alanı elde edilir.
 - Kayıt varsa, fakülte adı ilgili alana yazılır.
 - Kayıt bulunamazsa, hata mesajı verilir ve prosedür sonlandırılır.
- Yeni bölüm girişi değilse,
 - TBL_BOLUM tablosundan fakülteye kayıtlı, BOLUM = 1 olan tüm kayıtlar okunur.
 - Kayıt bulunamazsa, alanlar temizlenip prosedürden çıkılır.
 - Kayıt varsa, ad bilgisi ilgili alana yazılır.
 - Okunan tüm bölüm kodları kod combobox'ına yazılır.
 - İlk kayıt seçilir. (cmb_Bkod_Click çağırılır.)

cmb_Bkod_Click()

- Seçilen bölüm kodu boş değilse bu kod ile BolumDoldur prosedürü çağırılır.

bBYeni_Click()

- Fakülte combobox'ında kayıt varsa ilki seçilir. (cmb_BFakulte_Click çalışır.)
- Bölüm kodu ve başkan combobox'ları silinir.
- Ad alanı silinir.

bBKaydet_Click()

- Bölüm kodu, fakülte kodu alanları kontrol edilir. Boş ise hata verilerek prosedürden çıkılır.
- Seçilen bölüm kodu için TBL_BOLUM tablosu okunur. (BOLUM = 1 ile)
- Kayıt varsa,
 - Ekrandaki bilgiler ile UPDATE cümlesi oluşturulur. Boş olan alanlar NULL olarak alınır.
 - Kayıt güncellenir.
 - Kullanıcıya, işlemin bittiğini bildiren mesaj verilir.
- Kayıt yoksa,
 - Ekrandaki bilgiler ile INSERT cümlesi oluşturulur. Boş olan alanlar NULL olarak alınır. TBL_BOLUM tablosundaki BOLUM alanına ise '1' yazılır.
 - Kayıt yaratılır.
 - Kullanıcıya, işlemin bittiğini bildiren mesaj verilir.
 - BolumLoad prosedürü ile ekran tablodaki son bilgiler ile doldurulur.
 - Yeni eklenen kayıt BolumLoad prosedürü ile görüntülenir.

bBSil_Click()

- Bölüm kodu alanı kontrol edilir. Eğer boş ise hata verilerek prosedürden çıkılır.
- Verilen kod ile TBL_KULLANICI tablosu okunur. Bölüme tanımlı bir kullanıcı varsa bölüm kaydının silinmesine izin verilmez.
- Verilen kod ile BOLUM = 1 koşulu altında TBL_BOLUM tablosu okunur.

- Kayıt varsa,
 - Kullanıcının silme isteğini teyid etmesi gerekmektedir.
 - Kullanıcı “Evet” derse, ekrandaki bilgiler ile DELETE cümlesi oluşturulur. Kayıt silinir. Kullanıcıya işlemin bittiğini belirten mesaj verilir.
 - BolumLoad prosedürü çağırılır.
- Kayıt yoksa,
 - Kullanıcıya uyarı mesajı verilir.

Ana Bilim Dalı Tab’ı Fonksiyon ve Prosedürleri

Şekil 3.16’da ana bilim dallarının tanımlandığı bölüm gösterilmektedir.

ABDLoad()

- TBL_FAKULTE tablosundan tüm enstitüler FAKULTE = 0 kriteri altında okunur.
- Kayıt bulunamazsa tüm alanlar temizlenerek prosedür sonlandırılır.
- Okunan kayıtlar enstitü combobox’ına yazılır.
- Kayıt varsa ilki seçilir. (cmb_AFakulte_Click çalışır.)

ABDDoldur(pAKod As String)

- Kodu verilen ana bilim dalı için BOLUM = 0 olan kayıt TBL_BOLUM tablosundan okunur.
- Kayıt varsa ad alanı doldurulur. Ana bilim dalındaki profesörler başkan combobox’ına yazılır. Tablodan okunan başkan, bu combobox’dan seçilir.
- Kayıt yoksa, hata mesajı verilir. Alanlar temizlenir.

Şekil 3.16. Ana Bilim Dalı Tanımları

cmb_AFakulte_Click()

- Seçilen enstitü kodu boş değilse,
 - TBL_FAKULTE tablosu enstitü kodu ve FAKULTE = 0 için okunarak AD alanı elde edilir.
 - Kayıt varsa, enstitü adı ilgili alana yazılır.
 - Kayıt bulunamazsa, hata mesajı verilir ve prosedür sonlandırılır.
 - Yeni ana bilim dalı girişi değilse,

- TBL_BOLUM tablosundan enstitüye kayıtlı, BOLUM = 0 olan tüm kayıtlar okunur.
- Kayıt bulunamazsa, alanlar temizlenip prosedürden çıkılır.
- Kayıt varsa, ad bilgisi ilgili alana yazılır.
- Okunan tüm ana bilim dalı kodları kod combobox'ına yazılır.
- İlk kayıt seçilir. (cmb_AKod_Click çağırılır.)

cmb_AKod_Click()

- Seçilen ana bilim dalı kodu boş değilse bu kod ile ABDDoldur prosedürü çağırılır.

bAYeni_Click()

- Enstitü combobox'ında kayıt varsa ilki seçilir. (cmb_AFakulte_Click çalışır.)
- Ana bilim dalı kodu ve başkan combobox'ları silinir.
- Ad alanı silinir.

bAKaydet_Click()

- Ana bilim dalı kodu, enstitü kodu alanları kontrol edilir. Boş ise hata verilerek prosedürden çıkılır.
- Seçilen bölüm kodu için TBL_BOLUM tablosu BOLUM = 0 koşulu verilerek okunur.
- Kayıt varsa,
 - Ekrandaki bilgiler ile UPDATE cümlesi oluşturulur. Boş olan alanlar NULL olarak alınır.
 - Kayıt güncellenir.

- Kullanıcıya, işlemin bittiğini bildiren mesaj verilir.
- Kayıt yoksa,
 - Ekrandaki bilgiler ile INSERT cümlesi oluşturulur. Boş olan alanlar NULL olarak alınır. TBL_BOLUM tablosundaki BOLUM alanına ise sıfır yazılır.
 - Kayıt yaratılır.
 - Kullanıcıya, işlemin bittiğini bildiren mesaj verilir.
 - ABDLoad prosedürü ile ekran tablodaki son bilgiler ile doldurulur.
 - Yeni eklenen kayıt ABDLoad prosedürü ile görüntülenir.

bASil_Click()

- Ana bilim dalı kodu alanı kontrol edilir. Eğer boş ise hata verilerek prosedürden çıkılır.
- Verilen kod ile TBL_KULLANICI tablosu okunur. Ana bilim dalına tanımlı bir kullanıcı varsa ana bilim dalı kaydının silinmesine izin verilmez.
- Verilen kod ile BOLUM = 0 koşulu verilerek TBL_BOLUM tablosu okunur.
- Kayıt varsa,
 - Kullanıcının silme isteğini teyid etmesi gerekmektedir.
 - Kullanıcı “Evet” derse, ekrandaki bilgiler ile DELETE cümlesi oluşturulur. Kayıt silinir. Kullanıcıya işlemin bittiğini belirten mesaj verilir.
 - ABDLoad prosedürü çağırılır.
- Kayıt yoksa,
 - Kullanıcıya uyarı mesajı verilir.

3.6.4.6.Ders Tanımları (fDers)

Üniversitede verilen derslerin kaydedildiği, bilgilerinin değiştirildiği veya silindiği ekrandır. Ekran görüntüsü Şekil 3.17’de verilmiştir.

Açıklamalar

- Tip alanı, “LİSANS-SC”, “LİSANS-ZR”, “Y.LİSANS-SC”, “Y.LİSANS-ZR”, “DOKTORA-SC” ve “DOKTORA-ZR” değerlerinden birini alabilir.
- Yarıyıl alanı 1’den 8’e kadar değerler alabilir.

The screenshot shows a software window titled "Ders Tanımları". Inside the window, there are several text input fields arranged vertically. The labels for these fields are: "Tip", "Bölüm", "Ders", "Yarıyıl", "Öğretim Görevlisi", "Araştırma Görevlisi", "Haftalık Saat", "Kredi", and "Açıklama". Below these fields is a horizontal toolbar containing five buttons. From left to right, the buttons are: "Yeni Ders" (with a book icon), "Kaydet" (with a floppy disk icon), "Sil" (with a trash can icon), "Öğ Koşul Tanımlama" (with a warning triangle icon), and "Kapat" (with a close button icon).

Şekil 3.17. Ders Tanımları

Fonksiyon ve Prosedürler

bCik_Click()

- Ekran kapatılır. (Form_Unload çalışır.)

bKaydet_Click()

- Ders kodu alanı boş ise hata verilip prosedürden çıkılır.
- Verilen kod, tip ve bölüm için TBL_DERS tablosu okunur.
- Kayıt yoksa,
 - Ekrandaki bilgilerle INSERT cümlesi oluşturulur. Boş olan alanlar, veri tipine göre 0 veya NULL olarak alınır.
 - Kayıt yaratılır.
 - Kullanıcıya işlemin tamamlandığını belirten mesaj verilir.
 - Cmb_Bolum_Click çağırılır.
 - Yeni eklenen ders combobox'tan seçilir. (cmb_Kod_Click çalışır.)
- Kayıt varsa,
 - Ekrandaki bilgilerle UPDATE cümlesi oluşturulur. Boş olan alanlar, veri tipine göre 0 veya NULL olarak alınır
 - Kayıt güncellenir.
 - Kullanıcıya işlemin tamamlandığını belirten mesaj verilir.

bOn_Click()

- Ön Koşul Tanımlama ekranı çağırılır. Ders Tanımlama Ekranında görüntülenen ders varsa bu derse ait ön koşul tanımları ekrana getirilir.

bSil_Click()

- Ders kodu alanı boş ise hata verilip prosedürden çıkılır.
- Verilen kod, tip ve bölüm için TBL_DERS tablosu okunur.
- Kayıt varsa,
 - Kullanıcının silme isteğini teyid etmesi istenir.
 - “Evet” cevabı alınırsa, DELETE cümlesi oluşturulur. Kayıt silinir.
 - Kullanıcıya işlemin tamamlandığını belirten mesaj verilir.
 - EkranSil prosedürü çağırılır.
 - Cmb_Bolum_Click çağırılır.
- Kayıt yoksa,
 - Hata mesajı verilir.

bYeni_Click()

- EkranSil prosedürü çağırılır.
- Kod combobox'ı silinir.

cmb_Bolum_Click()

- Bölüm kodu boşluk değilse,
 - EkranSil prosedürü çağırılır.
 - Seçilen bölüm kodu ile TBL_BOLUM tablosundaki AD alanı okunur.
 - Kayıt varsa ilgili alana yazılır, yoksa ekrandaki ad alanı silinir.

- Bölümde görev yapan ve ünvanı “Araştırma Görevlisi”nden sonra olan kullanıcılar TBL_KULLANICI tablosundan okunur. Bulunan kayıtlar ilgili combobox’a yazılır. Kayıt varsa ilki seçilir.
- Bölümde görev yapan ve ünvanı “Araştırma Görevlisi” olan öğretim üyeleri TBL_KULLANICI tablosundan okunur. Bulunan kayıtlar ilgili combobox’a yazılır. Kayıt varsa ilki seçilir.
- Ekranda seçilen tip ve bölüme göre TBL_DERS tablosu okunur.
- Okunan dersler combobox’a yazılır. Varsa ilk kayıt silinir. (cmb_Kod_Click çağırılır.)
- Bölüm kodu boş ise,
 - Alanlar temizlenir.

cmb_Kod_Click()

- Seçilen ders kodu boşluk değilse bu kod ile EkranDoldur prosedürü çağırılır.

cmb_Tip_Click()

- Seçilen tip boşluktan farklıysa,
 - EkranSil prosedürü çağırılır.
 - Ders tipi, yüksek lisans veya doktora ise TBL_BOLUM tablosundan BOLUM = 0 ile ana bilim dalları okunur.
 - Ders tipi, hazırlık veya lisans ise TBL_BOLUM tablosundan BOLUM = 1 bölümler okunur.
 - Okunan kayıtlar bölüm combobox’ına yazılır. Kayıt varsa, ilk kayıttaki ad ilgili alana yazılır.
- Seçilen ders tipi boşluk ise,

- Alanlar silinir.

Form_Load()

- Global bDers değişkenine TRUE değeri atanır.

Form_Unload(Cancel As Integer)

- Global bDers değişkenine FALSE değeri atanır.

EkranDoldur()

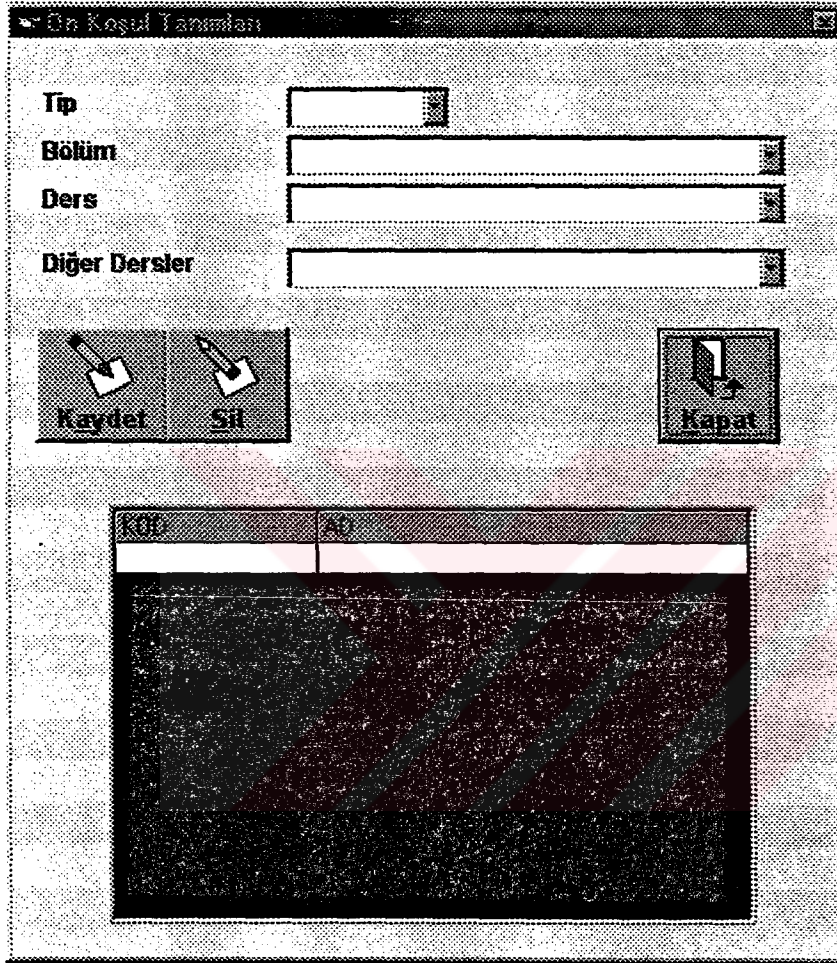
- EkranSil prosedürü çağırılır.
- Seçilen ders kodu, tip, bölüm kodu için TBL_DERS tablosu okunur.
- Kayıt bulunamazsa hata mesajı verilerek prosedür sonlandırılır.
- Kayıttaki bilgiler ile ekrandaki alanlar doldurulur.

EkranSil()

- Ekrandaki alanlar temizlenir.

3.6.4.7.Ön Koşul Dersleri Tanımlama (fOn)

Belirli bir derse kayıt olabilmek için, daha önceki dönemlerde alınmış olması gereken ön koşul derslerinin tanımlandığı ekrandır. Şekil 3.18’de ekran görüntüsü verilmiştir.



KOD	AD
-----	----

Şekil 3.18. Ön Koşul Tanımlama

Açıklamalar

- Tip alanı, “LİSANS-SC”, “LİSANS-ZR”, “Y.LİSANS-SC”, “Y.LİSANS-ZR”, “DOKTORA-SC” ve “DOKTORA-ZR” değerlerinden birini alabilir.

Fonksiyon ve Prosedürler

bCik_Click()

- Ekran kapatılır. (Form_Unload çalışır.)

bKaydet_Click()

- Seçilen ders ve ön koşul dersi kodu ile TBL_ONKOSUL tablosu okunur.
- Kayıt yoksa,
 - Ekranda seçilen ders ve ön koşul ders bilgileri ile INSERT cümlesi oluşturulur.
 - Kayıt yaratılır.
 - Kullanıcıya işlemin tamamlandığı bildirilir.
 - Cmb_Kod_Click prosedürü çağırılarak grid'in güncellenmesi sağlanır.
 - Yeni tanımlanan koşul dersi, diğer dersler combobox'ında seçilir.
- Kayıt varsa,
 - Kullanıcıya dersin zaten ön koşul olarak tanımlı olduğu bildirilir.

bSil_Click()

- Seçilen ders ve ön koşul dersi kodu ile TBL_ONKOSUL tablosu okunur.
- Kayıt varsa,
 - Ekranda seçilen ders ve ön koşul ders bilgileri ile DELETE cümlesi oluşturulur.
 - Kayıt silinir.
 - Kullanıcıya işlemin tamamlandığı bildirilir.
 - Cmb_Kod_Click prosedürü çağırılarak grid'in güncellenmesi sağlanır.

- Kayıt yoksa,
 - Kullanıcıya hata mesajı verilir.

cmb_Bolum_Click()

- Boşluk dışında bir bölüm kodu seçilmişse,
 - EkranSil prosedürü çağırılır.
 - Kayıt bulunursa, ilgili alana yazılır. Yoksa ilgili alan silinir.
 - Seçilen bölüme ait dersler TBL_DERS tablosundan okunur.
 - Okunan kayıtlar ders combobox'ına yazılır. Kayıt varsa ilki seçilir.
- Seçilen bölüm kodu boş ise,
 - Bölüm ve ders ile ilgili alanlar temizlenir.

cmb_Kod_Click()

- Seçilen ders kodu boşluk değilse, bu kod ile EkranDoldur prosedürü çağırılır.

cmb_Tip_Click()

- Seçilen tip kodu boşluk değilse,
 - Seçilen tip yüksek lisans veya doktora ise TBL_BOLUM tablosundaki ana bilim dallarının BOLUMNO ve AD alanları BOLUM = 0 koşuluyla okunur.
 - Seçilen tip hazırlık veya lisans ise TBL_BOLUM tablosundaki bölümlerin BOLUMNO ve AD alanları BOLUM = 1 şartıyla okunur.
 - Elde edilen kayıtlar bölüm combobox'ına yazılır. Varsa ilki seçilir. (cmb_Bolum_Click çalışır.)
- Seçilen tip kodu boşluk ise,

- Bölüm ile ilgili alanlar silinir.

EkranDoldur(tKod As String)

- EkranSil prosedürü çağırılır.
- TBL_DERS tablosundan seçilen tip ve bölüm için tanımlanmış olan ve kodu verilen parametreden farklı olan tüm dersler okunur.
- Varsa ilk kayıttaki ad, ön koşul dersi adı alanına yazılır.
- Bulunan ders kodları ön koşul dersi combobox'ına yazılır. Varsa ilk kayıt seçilir. (cmb_KosulDers_Click çalışır.)
- Parametre ile verilen ders koduna tanımlı ön koşul derslerin adları TBL_ONKOSUL ve TBL_DERS tablolarının join ile okunmasıyla bulunur.
- Kayıt varsa,
 - Grid silinir. Satır sayısı, bulunan kayıt sayısının 2 fazlası olarak belirlenir.
 - MSFGFormatla prosedürü çağırılır.
 - Bulunan kayıtlar gride aktarılır.

EkranSil()

- Ön koşul alanları ve grid silinir.
- Grid'in satır sayısı 2 olarak verilir.
- MSFGFormatla prosedürü çağırılır.

Form_Load()

- MSFGFormatla prosedürü çağırılır.
- Ders Tanımları ekranından bir ders seçilene gelinmişse,

- FDers formundaki bilgiler bu ekrana aktarılır. (cmb_Tip_Click, cmb_Bolum_Click, cmb_Kod_Click, cmb_KosulDers_Click çalışır.)

MSFGFormatla()

- Grid'in kolon genişlikleri verilir.
- Başlıklar yazılır



3.6.4.8.Ders Programı Tanımları (fDersPrg)

Şekil 3.19’da görülen ekranda ders programlarının tanımları yapılmaktadır.

	PAZARTESİ	SALI	ÇARŞAMBA	PERŞEMBE	CUMA
09:00-09:50					
10:00-10:50					
11:00-11:50					
12:00-12:50					
13:00-13:50					
14:00-14:50					
15:00-15:50					
16:00-16:50					
17:00-17:50					

Şekil 3.19. Ders Programı Tanımları

Açıklamalar

- Gün alanı, “Pazartesi”, “Salı”, “Çarşamba”, “Perşembe”, “Cuma” değerlerinden birini alabilir.

- Seans alanı, “09:00-09:50”, “10:00-10:50”, “11:00-11:50”, “12:00-12:50”, “13:00-13:50”, “14:00-14:50”, “15:00-15:50”, “16:00-16:50”, ‘17:00-17:50’ değerlerinden birini alabilir.
- Tip alanı, “LİSANS-SC”, “LİSANS-ZR”, “Y.LİSANS-SC”, “Y.LİSANS-ZR”, “DOKTORA-SC” ve “DOKTORA-ZR” değerlerinden birini alabilir.

Fonksiyon ve Prosedürler

bCik_Click()

- Ekran kapatılır. (Form_Unload çalışır.)

bKaydet_Click()

- Bina ve sınıf alanları kontrol edilir. Boş ise hata verilerek prosedür sonlandırılır.
- TBL_DERSPRG tablosu verilen gün, seans, ders, yıl ve yarıyıl için okunur. Kayıt varsa, kullanıcıya "Belirttiğiniz Kayıt Daha Önce Tanımlanmış! Değiştirmek İstiyorsanız Önce Siliniz!" mesajı verilir. Prosedür sonlandırılır.
- TBL_DERSPRG tablosu verilen gün, seans, bina, sınıf, ders, yıl ve yarıyıl ile okunur. Seçilen dersten başka bir ders kayıtlı ise kullanıcıya "Belirttiğiniz Saatte Sınıf Doludur!" mesajı verilir. Prosedür sonlandırılır.
- Aynı gün ve seansta aynı bölüm ve yarıyılın dersi olup olmadığına bakılır. (TBL_DERSPRG ve TBL_DERS). Kayıt varsa, kullanıcıya " Belirttiğiniz Saatte Aynı Bölüm ve Yarıyılın Başka Bir Dersi Var. Lütfen Kontrol Ediniz!" mesajı verilir. Bu durumda proses sonlandırılmaz
- Seçilen ders için TBL_DERS'teki toplam haftalık saat bilgisi okunur. TBL_DERSPRG dosyasındaki toplam saat bu değeri aşıyorsa kullanıcıya "Dersin Haftalık Toplam Saati Dolmuştur!" mesajı verilir. Prosedür sonlandırılır.
- Kontrollerin ardından, INSERT cümlesi oluşturulur ve kayıt yaratılır.

- Kullanıcıya işlemin sonlandığı bildirilir.
- Yeni yaratılan kayıt, grid'e yazılır.

bSil_Click()

- TBL_DERSPRG tablosu verilen gün, seans ve ders için okunur.
- Kayıt yoksa hata verilip prosedürden çıkılır.
- DELETE cümlesi oluşturulup kayıt TBL_DERSPRG tablosundan silinir.
- Kullanıcıya silme işleminin bitirildiği bildirilir.
- Ders combobox'ında kayıt varsa cmb_Ders_Click çağırılır.

cmb_Bina_Click()

- TBL_SINIF tablosundan seçilen binaya tanımlı sınıflar okunur.
- Okunan kayıtlar sınıf combobox'ına yazılır.
- Varsa ilk kayıt seçilir.

cmb_Bolum_Click()

- Seçilen bölüm için TBL_DERS tablosundaki DERSNO ve AD alanları okunur.
- Elde edilen kayıtlardaki ad alanı ders combobox'ına yazılır. Yazım sırasında combobox'ın ItemData özelliğine de, yazılan dersin kodu aktarılır.
- Kayıt varsa ilki seçilir. (cmb_Ders_Click çalışır.)

cmb_Ders_Click()

- Seçilen derse ait dönem bilgisi TBL_DERS tablosundan okunur ve ekrandaki ilgili alana yazılır.

- Seçilen ders ile aynı dönemde verilen bölüm derslerine ait TBL_DERSPRG kayıtları okunur.
- Grid temizlenir.
- MSFGFormatla prosedürü çağırılır.
- Okunan kayıtlar griddeki uygun yerlerine yerleştirilir.

cmb_Tip_Click()

- Seçilen tip kodu boşluk değilse,
 - Seçilen ders tipi yüksek lisans veya doktora ise TBL_BOLUM tablosundaki ana bilim dallarının BOLUMNO ve AD alanları BOLUM = 0 koşuluyla okunur.
 - Seçilen tip hazırlık veya lisans ise TBL_BOLUM tablosundaki bölümlerin BOLUMNO ve AD alanları BOLUM = 1 şartıyla okunur.
 - Elde edilen kayıtlar bölüm combobox'ına yazılır. Varsa ilki seçilir. (cmb_Bolum_Click çalışır.)
- Seçilen tip kodu boşluk ise,
 - Bölüm ile ilgili alanlar silinir.

cmb_Yıl_Click()

- Geçerli bir yıl seçilmişse cmb_Ders_click çalıştırılır.

Form_Load()

- Gün, seans ve ders tipi alanları SabitDoldur prosedürü çağırılarak ilgili bilgiler ile doldurulur.
- TBL_BINA tablosundan KOD, AD alanları okunur.
- Okunan kayıtlar bina combobox'ına yazılır.

- Varsa ilk bina kaydı seçilir. (cmb_Bina_Click çalışır.)
- Grid temizlenir.
- MSFGFormatla prosedürü çağırılır.

MSFGFormatla()

- Satır yüksekliği ve kolon genişliği verilir.
- İlk satıra günler, ilk sütuna seanslar yazılır.



3.6.4.9.Ders Notu Girişi (fNot)

Öğretim üyelerinin veya öğrenci işleri yetkililerinin ders notlarını girdikleri ekrandır.

Şekil 3.20’de ekran görüntüsü verilmiştir.

Hak	Katılım Saat
Vize 1	Quiz 1
Vize 2	Quiz 2
Final	Ödev 1
	Ödev 2
Sayısal Ortalama	Ödev 3
Dönem Sonu	Ödev 4

Şekil 3.20. Not Girişi

Açıklamalar

- Hak ve katılım saat alanları değiştirilemez.
- Dönem sonu ortalama alanı "AA", "BA", "BB", "CB", "CC", "DC", "DD", "F" değerlerinden birini alabilir.

Fonksiyon ve Prosedürler

bCik_Click()

- Ekran kapatılır. (Form_Unload çalışır.)

bGetir_Click()

- Öğretim üyesinin kullanıcı numarası girilmemişse hata mesajı verilir, prosedür sonlandırılır.
- EkranSil prosedürü çağırılır.
- Kullanıcı numarası verilen öğretim üyesinin adı, soyadı TBL_KULLANICI tablosundan okunur.
- Kayıt bulunursa, ilgili alanlar doldurulur. Yoksa, kullanıcıya hata verilir ve prosedür sonlandırılır.
- Öğretim üyesinin verdiği derslerin kod ve adı TBL_DERS tablosundan okunur.
- Okunan kayıtlar ders combobox'ına yazılır.
- Varsa ilk kayıt seçilir. (cmb_Ders_Click çalışır.)

bKaydet_Click()

- Yıl, dönem, ders, öğrenci numarası alanları kontrol edilir. Eğer herhangi biri boş geçilmişse hata mesajı verilip prosedürden çıkılır.
- Girilen kullanıcı numarası ile TBL_KULLANICI tablosu okunur. Kayıt bulunamazsa hata verilip prosedürden çıkılır.
- Seçilen ders kodu ile TBL_DERS tablosu okunur. Kayıt yoksa hata verilir. Prosedür sonlandırılır.
- Seçilen yıl, dönem, öğrenci ve ders kodu ile TBL_OGRDERS tablosu okunur. Kayıt yoksa hata verilip çıkılır.

- Ekrandaki bilgiler ile UPDATE cümlesi oluşturulur. Boş geçilen alanlar NULL olarak alınır.
- Kayıt güncellenir.
- Kullanıcıya işlemin tamamlandığını belirten mesaj verilir.

bOku_Click()

- Giriş ekranında yer alan GemPlus410 kontrolünün 'Oku' metodu çağırılır.
- Kontrolün ilgili özellikleri kontrol edilir, hata oluşmuşsa prosedürden çıkılır.
- Okuma işlemi başarı ile sonuçlanmışsa karttan okunan kullanıcı numarası ekrandaki ilgili alana yazılır.

cmb_Ders_Click()

- EkranSil prosedürü çağırılır.
- TBL_OGRDERS tablosundan seçilen yıl ve dönemde dersi alan öğrenciler okunur.
- Elde edilen kayıtlar öğrenci combobox'ına yazılır.
- Varsa ilk kayıt seçilir. (cmb_No_Click çalışır.)

cmb_No_Click()

- EkranSil prosedürü çağırılır.
- Seçilen numara ile TBL_KULLANICI tablosundaki AD, SOYAD alanları okunur.
- Kayıt varsa,
 - Ad, soyad birleştirilerek ilgili alana yazılır.
 - Öğrencinin bilgileri TBL_OGRDERS tablosundan okunur.

- Tablodaki deęerler ekrandaki ilgili alanlara yazılır. NULL olan alanlar boşluk olarak yazılır.
- Kayıt yoksa,
 - Hata verilerek prosedür sonlandırılır.

cmb_Yıl_Click()

- Seçili ders kodu varsa cmb_Ders_Click çağırılır.

Form_Load()

- TBL_PARAM tablosu okunur.
- Okunan kayıttaki YIL alanı ilgili combobox'a yazılır..

opt_Kis_Click()

- Seçili ders kodu varsa cmb_Ders_Click çağırılır.

opt_Yaz_Click()

- Seçili ders kodu varsa cmb_Ders_Click çağırılır.

Ekransil()

- Ekrandaki text alanlar silinir.
- Ortalama combobox'ından ilk kayıt seçilir.

3.6.4.10. Parametre Girişi (fParam)

Şekil 3.21’de görüntüsü verilen bu ekranda dönem başlangıç–bitiş tarihi gibi parametreler sisteme girilir, değiştirilir.

Şekil 3.21. Parametre Girişi

Fonksiyon ve Prosedürler

bCik_Click()

- Ekran kapatılır. (Form_Unload çalışır.)

bKaydet_Click()

- Kayıt başlangıç tarihi bitiş tarihinden sonraysa hata verilir ve prosedür sonlandırılır.
- Dönem başlangıç tarihi bitiş tarihinden sonraysa hata verilir ve prosedür sonlandırılır.
- Final başlangıç tarihi bitiş tarihinden sonraysa hata verilir ve prosedür sonlandırılır.
- Kayıt başlangıç tarihi dönem başlangıç tarihinden sonraysa hata verilir ve prosedür sonlandırılır.

- Dönem başlangıç tarihi final başlangıç tarihinden sonraysa hata verilir ve prosedür sonlandırılır.
- TBL_PARAM tablosundan seçilen yıl ve dönem için kayıt okunur.
- Kayıt varsa,
 - UPDATE cümlesi oluşturulur. Boş olan alanlar NULL olarak alınır.
 - Kayıt güncellenir.
 - Kullanıcıya işlemin tamamlandığına ilişkin mesaj verilir.
- Kayıt yoksa,
 - INSERT cümlesi oluşturulur. Boş olan alanlar NULL olarak alınır.
 - Kayıt yaratılır.
 - Kullanıcıya işlemin tamamlandığına ilişkin mesaj verilir.

cmb_Yil_Click()

- Seçilen yıl boşluk değilse,
 - TBL_PARAM tablosundan o yıl ve seçili döneme ait kayıt okunur.
 - Kayıt varsa bilgiler ekrana yazılır.
 - Kayıt yoksa EkranSil prosedürü çağırılır

Form_Load()

- Yıl combobox'ına 2000 – 2030 değerleri yazılır. İlki seçilir. (cmb_Yil_Click çalışır.)
- Yaz dönemi seçilir.

EkranSil()

- Text alanlar temizlenir.

opt_Kis_Click()

- cmb_Yil_Click prosedürü çağırılır.

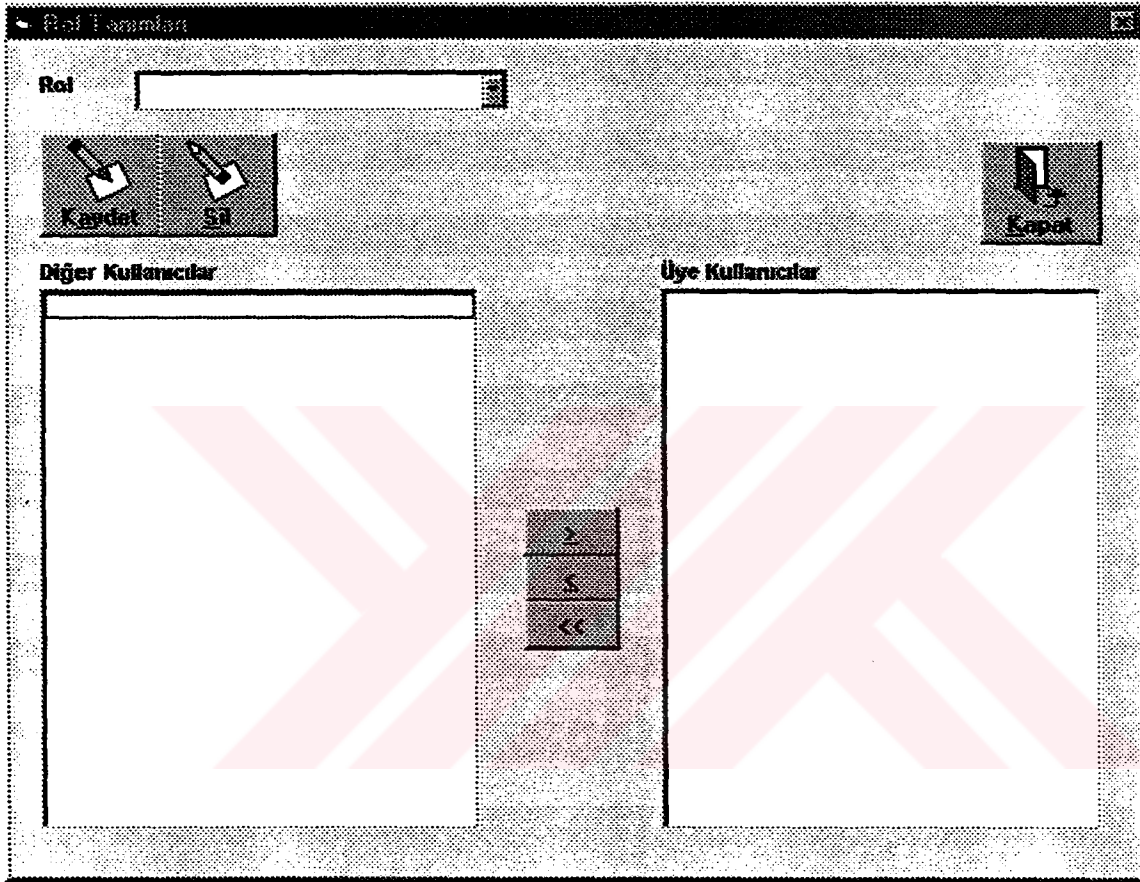
opt_Yaz_Click()

- cmb_Yil_Click prosedürü çağırılır.



3.6.4.11. Rol Tanımları

İTÜKart Yönetim Otomasyonu içindeki ekran yetkilerinin verildiği roller Şekil 3.22’de gösterilen ekran üzerinden tanımlanır. Sistem kullanıcılarına rolleri yine bu ekran verilir.



Şekil 3.22. Rol Tanımları

Fonksiyon ve Prosedürler

bCik_Click()

- Ekran kapatılır. (Form_Unload çalışır.)

b_Cikar_Click()

- Rol üyeleri listesinde seçili olan kullanıcı(lar) ekranın solunda yer alan 'Diğer Kullanıcılar' listesine aktarılır.

b_Ekle_Click()

- 'Diğer Kullanıcılar' listesinde seçili olan kullanıcı(lar) 'Üye Kullanıcılar' listesine aktarılır.

b_HepsiCikar_Click()

- 'Üye Kullanıcılar' listesindeki tüm kayıtlar 'Diğer Kullanıcılar' listesine aktarılır.

b_Kaydet_Click()

- Rol alanı boş ise kullanıcıya "Lütfen Rol Belirtiniz!" hatası verilerek prosedür sonlandırılır.
- Ekranda verilen rol ile TBL_ROLE tablosu okunur.
- Kayıt varsa mevcut kayıtlar silinir.
- 'Üye kullanıcılar' listesindeki kullanıcılar ve rol için TBL_ROLE tablosunda yeni kayıtlar yaratılır.
- FillRol prosedürü çağırılarak ekran yenilenir.

b_Sil_Click()

- Belirtilen rolün TBL_ROLE tablosundaki kayıtlarının sayısı okunur.
- Eğer kayıt bulunamazsa kullanıcıya hata verilir ve prosedür sonlandırılır.
- TBL_ROLE tablosunda rol için tanımlı bulunan kayıtlar silinir.
- FillRol prosedürü çağırılarak ekran yeniden doldurulur.

cmb_Rol_Click()

- Ekrandaki listeler temizlenir.
- Role tanımlı olmayan kullanıcılar okunur ve 'Diğer Kullanıcılar' listesine yazılır.
- Role tanımlı olan kullanıcılar okunur ve 'Üye Kullanıcılar' listesine yazılır.

Form_Load()

- FillRol prosedürü çağırılarak ekran yeniden doldurulur.

FillRol()

- Rol combobox'ı temizlenir.
- TBL_ROL tablosundaki rol değerleri okunur ve rol combobox'ına yazılır.
- . Listedeki ilk rol seçilir. (cmb_rol_click çalışır.)

lst_Diger_DblClick()

- b_Ekle_Click prosedürü çağırılarak seçilen kayıt üye kullanıcılar listesine aktarılır.

lst_Uye_DblClick()

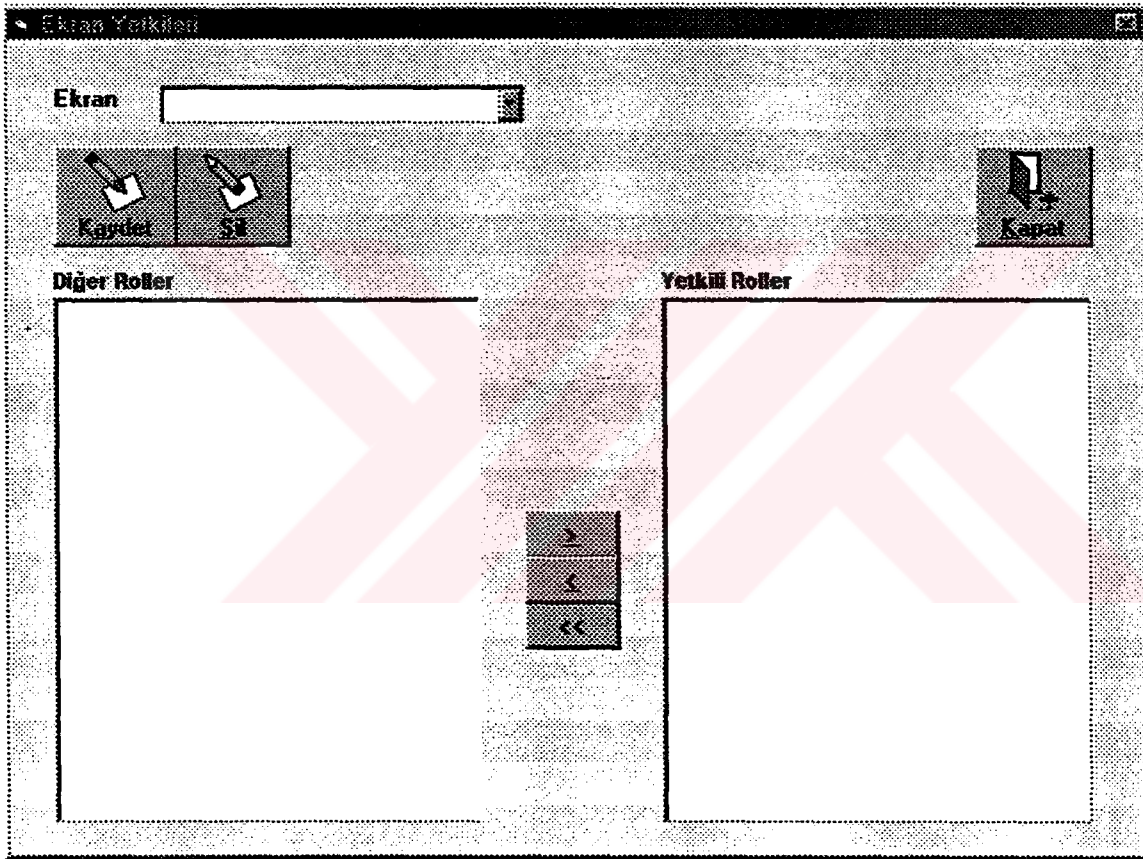
- b_Cikar_Click prosedürü çağırılarak seçilen kayıt diğer kullanıcılar listesine aktarılır.

3.6.4.12. Ekran Yetkileri

İTÜKart Yönetim Otomasyonu'nda yer alan ekranlar için sistemdeki ilgili rollere giriş yetkisinin verildiği bu ekranın görüntüsü Şekil 3.23'te verilmiştir.

Açıklama

- Yetkilendirme işlemi sırasında ekran isimleri olarak ilgili Visual Basic formunun ismi verilir. Bu, dosya ismiyle aynı olmak zorunda değildir.



Şekil 3.23. Ekran Yetkileri

Fonksiyon ve Prosedürler

bCik_Click()

- Ekran kapatılır. (Form_Unload çalışır.)

b_Cikar_Click()

- ‘Yetkili Roller’ listesinde seçili olan rol(ler) bu listeden çıkarılıp ‘Diğer Roller’ listesine eklenir.

b_Ekle_Click()

- ‘Diğer Roller’ listesinde seçili olan rol(ler) bu listeden çıkarılıp ‘Yetkili Roller’ listesine eklenir.

b_HepsiCikar_Click()

- ‘Yetkili Roller’ listesinde yer alan tüm roller bu listeden çıkarılıp ‘Diğer Roller’ listesine eklenir.

b_Kaydet_Click()

- Ekran ismi belirtilmemişse hata verilerek prosedürden çıkılır.
- TBL_ROLEKRAN tablosundan seçili ekrana giriş yetkisi bulunan roller okunur.
- Bulunan kayıtlar silinir.
- ‘Yetkili Roller’ listesindeki roller, seçili ekran için TBL_ROLEKRAN’a eklenir.
- FillEkran prosedürü çağırılarak ekrandaki alanlar yeniden doldurulur.

b_Sil_Click()

- TBL_ROLEKRAN tablosundan seçili ekran için yetkisi bulunan rollerin sayısı okunur.
- Kayıt bulunamazsa hata verilerek prosedür sonlandırılır.
- TBL_ROLEKRAN tablosundan ekran ile ilgili kayıtlar silinir.
- FillEkran prosedürü çağırılarak ekrandaki alanlar yeniden doldurulur.

cmb_ekran_Click()

- Ekran için yetkisi bulunmayan roller okunarak ‘Diğer Roller’ listesine yazılır.

- Ekran için giriş yetkisine sahip olan roller okunarak ‘Yetkili Roller’ listesine yazılır.

Form_Load()

- FillEkran prosedürü çağırılarak ekrandaki alanlar doldurulur.

FillEkran()

- TBL_ROL tablosundan sistemde tanımlı olan roller okunarak ‘Diğer Roller’ listesine yazılır.
- Ekran isimlerinin bulunduğu combobox temizlenir.
- TBL_ROLEKRAN tablosundan sistemde bulunan ekran isimleri okunarak ilgili alana yazılır. Sistemde tanımlı ekran yoksa hata verilerek çıkılır.
- İlk ekran seçilir. (cmb_Ekran_Click çalışır.)

lst_Diger_DblClick()

- b_Ekle_Click prosedürü çağırılarak listede seçili olan rol ‘Yetkili Roller’ listesine aktarılır.

lst_Uye_DblClick()

- b_Cikar_Click prosedürü çağırılarak listede seçili olan rol ‘Diğer Roller’ listesine aktarılır.

SONUÇ

Tez çalışmasının başında, standart bir kampüs kartının temel işlevlerini yerine getiren bir prototip hazırlanması amaçlanmıştır. Bu doğrultuda İTÜKart sistemi planlanan şekilde geliştirilmiştir.

Proje kapsamında tasarlanan veri tabanı MS SQL Server 7.0 üzerinde hayata geçirilmiştir. Uygulamaya dinamik bir yapı kazandırılması amacıyla programlarda kullanılan sabit değerler mümkün olduğunca tablolarda tutulmuştur. Performansın düşürülmemesi için tasarım, sadece prototip işlevlerinin gereksinimleri gözönüne alınarak yapılmıştır. Daha sonraki aşamalarda yeni özelliklerin eklenmesi sırasında veritabanı tasarımı da tekrar gözden geçirilmelidir.

Uygulamada kullanılacak teknoloji ve araç seçimi aşamasında piyasadaki ürünler incelenmiş, sunduğu yüksek performans/maliyet oranı, kolay ve etkin uygulama geliştirme imkanları (hazır kütüphane fonksiyonları, kart üzerindeki dosya yapısıyla ilgili işlemler için geliştirilmiş GemPilot Yönetim modülü) nedeniyle GemPlus firmasının GemClub ürününün kullanılmasına karar verilmiştir. Programlar ile kart okuyucu arasındaki iletişimin sağlanabilmesi için GemClub kütüphane fonksiyonlarının kullanıldığı arayüz ActiveX teknolojileriyle Visual Basic 6.0 kullanılarak yazılmıştır. ActiveX kontrol olarak hazırlanan genel amaçlı arayüz programı ileride benzer uygulamalarda da rahatlıkla kullanılabilir.

Sistemin çalışması için gerekli tanımlamaların yapıldığı İTÜKart Yönetim Otomasyonu Visual Basic 6.0 ile çalıştırılmaya hazır “exe” uzantılı olarak hazırlanmıştır. Bu sayede kurulum paketi kolayca hazırlanıp yetkili kullanıcıların bilgisayarlarına rahatlıkla kurulabilir. Program içinde veritabanına yapılan bağlantılar nesneye dayalı yaklaşım ile aynı bir ActiveX nesnesi üzerinden sağlanmaktadır. Böylece program kodunda veritabanı bilgileri görülemediğinden veri güvenliği arttırılmaktadır [1].

İTÜKart Kullanıcı Otomasyonu web ortamında çalışacak şekilde gerçekleştirilmiştir. İTÜKart Yönetim Otomasyonu ile ortak altyapı üzerinde çalışan bu proram sayesinde sistem kullanıcılarının ders, randevu ve ödeme işlemlerini sahip oldukları kartları da kullanarak yapabilmeleri sağlanmıştır.[1].

Geliştirilen İTÜKart sistemi sayesinde daha önce de kısmen yapılabilen internet üzerinden ders seçimi, ders notlarının sorgulanması gibi işlemler geliştirilmiş ve daha güvenli bir şekilde gerçeklenmeleri sağlanmıştır. Bunun yanı sıra sisteme mevcut yapıda bulunmayan randevu, ödeme gibi yeni işlevler sisteme kazandırılmıştır. Ayrıca, daha önce sadece kimlik kartı olarak görev yapan akıllı kart, bu proje sayesinde diğer işlemler için de kullanılarak sistemin tüm olanaklarından tek bir kart ile faydalanılması sağlanmıştır [1].

Hazırlanan bu prototip esas alınarak daha fazla işleve sahip bir üniversite kampüs kartı sistemi hazırlanabilir.

- Veritabanında tutulan kullanıcı şifresi açık olarak değil, şifrelenmiş şekilde saklanmaktadır. İleride bu şifreleme algoritması geliştirilebilir ya da dünyada kabul görmüş bir algoritma (DES, DES3, vb) uyarlanabilir.
- Güvenliğin artırılması amacı ile karttaki kayıt dosyaları farklı işlemler için (okuma, güncelleme, vb) farklı şifreler ile korunabilir.
- Prototip'e Kütüphane Otomasyonu, Mediko-Sosyal Otomasyonu gibi yeni modüller eklenebilir, mevcut modüller genişletilebilir. Örneğin İTÜKart Kullanıcı Otomasyonu'nda öğrencinin sınav programını görebileceği bir sayfa hazırlanabilir.
- Mevcut prototipte kart üzerinde sadece kullanıcı numarası ve kredi miktarı tutulmaktadır. Daha sonraki aşamalarda kullanıcıya ait diğer bilgiler de (kimlik bilgileri, kütüphaneden alınan kitaplara ilişkin takip bilgileri,vs) kart üzerinde depolanabilir.

- Mevcut sistem bir prototip olduğundan bazı işlemler basitleştirilmiştir. Örneğin, öğrenciler için paket ders seçeneğinin bulunmadığı varsayılmıştır. Bu gibi varsayımlar ileriki versiyonlarda ortadan kaldırılarak program geliştirilebilir.
- Üzerinde analizler yapılabilecek istatistiklerin alınabileceği Raporlama modülü eklenebilir.
- İTÜKart Yönetim Otomasyonu'nda kurum tanımlama, kartlara kredilerin yüklenebileceği banka arayüzü gibi yeni modüller hazırlanabilir.
- Mevcut Yönetim Otomasyonu sayfalarına kullanıcı memnuniyetini arttırmak üzere 'Yardım' seçeneği eklenebilir.
- Yönetim Otomasyonu'nda kullanım kolaylığı sağlamak amacıyla ders programı tanımlama, sınav programı tanımlama gibi ekranlara yazıcıdan çıktı alma işlevi gerçekleştirilebilir.

Benzer uygulamaların artmasıyla birlikte ülkemizde de akıllı kart kullanım oranı yakın bir gelecekte artacaktır. Hali hazırda devam etmekte olan MERNİS – Ulusal Kimlik Kartı, SSK – Bağkur projeleri tamamlandığında bu sistemlerin Emniyet Müdürlüğü ve Türk Telekom sistemleriyle tümleştirilmesi sonucunda Türkiye Cumhuriyeti vatandaşlarının tek bir kart ile kamu kurumlarına ait kaynakları kullanmaları, tüm resmi işlemleri gerçekleştirmeleri ve takip etmeleri sağlanacaktır.

KAYNAKLAR

- [1] **ÇÖLOĞLU, Ayşe E.**, 2001. İTÜKart Kullanıcı Otomasyonu, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [2] **ISO 7816-1**, 1998. Design and Use of Identification Cards Having Integrated Circuit with Contacts - Physical Characteristics, *International Standarts Organization*, Cenevre.
- [3] **Card Europe UK** (1994). *Smart Card Technology Background Paper*, <<http://www.cardeurope.demon.co.uk/repl.htm>> adresli internet sitesi (27 Mayıs 1999)
- [4] **Smart Card Technology** (1998). *Smart Card Technology*, <<http://www.oberthurkirk.com/technology.htm>> adresli internet sitesi (06 Eylül 1998)
- [5] **Purdue University – Smart Card Page** (1996). *Smart Cards*, <<http://www.tech.purdue.edu/it/resources/aidc/smcard.htm>> adresli internet sitesi (07 Mayıs 1999)
- [6] **Global Electronic Commerce** (1998). *Smart Card with Electronic Commerce*, <<http://ubmail.ubalt.edu/~kkim/smartcard.htm>> adresli internet sitesi (24 Şubat 2000)
- [7] **ISO 7816-2**, 1999. Design and Use of Identification Cards Having Integrated Circuit with Contacts – Contact Locations and Minimum Size, *International Standarts Organization*, Cenevre.
- [8] **ISO 10536**, 1987. Identification cards -- Contactless integrated circuit(s) cards, *International Standarts Organization*, Cenevre.
- [9] **Smart Card News: Introduction to Smart Cards** (1998). *Smart Card Technology: Introduction to Smart Cards*, <<http://www.smartcard.co.uk>> adresli internet sitesi (06 Eylül 1998)
- [10] **ISO 7816-4**, 1995. Design and Use of Identification Cards Having Integrated Circuit with Contacts – Inter-Industry Commands for Interchange, *International Standarts Organization*, Cenevre.

- [11] **ISO 7816-5**, 1994. Design and Use of Identification Cards Having Integrated Circuit with Contacts – Application Identification, *International Standards Organization*, Cenevre.
- [12] **ISO 7816-6**, 1996. Design and Use of Identification Cards Having Integrated Circuit with Contacts – Inter-Industry Data Elements, *International Standards Organization*, Cenevre.
- [13] **Smart Card and Security Overview** (2000). *Smart Card and Security Overview*, <<http://www.smartcardbasic.com>> adresli internet sitesi (22 Ekim 2000)
- [14] **ISO 7816-7**, 1999. Design and Use of Identification Cards Having Integrated Circuit with Contacts – Interindustry commands for Structured Card Query Language (SCQL), *International Standards Organization*, Cenevre.
- [15] **An Overview of Smart Card Security** (1999). *An Overview of Smart Card Security*, <<http://www.hkstar.com/~alanchan/papers/smartcardsecurity>> adresli internet sitesi (22 Ekim 2000)
- [16] **ISO 7816-3**, 1997. Design and Use of Identification Cards Having Integrated Circuit with Contacts – Electrical Specifications and Communication Protocols, *International Standards Organization*, Cenevre.
- [17] **ISO 7810 (ID-1)**, 1995. Identification cards - Physical characteristics, *International Standards Organization*, Cenevre.
- [18] **Smart Card Standards and electronic Purse** (1997). *Smart Card Standards and Electronic Purse*, <<http://www.aston.ac.uk/smartcard/documentation/standards1.hmt>> adresli internet sitesi (24 Şubat 2000)
- [19] **Smart Card Applications** (1998). *Smart Card Applications*, <<http://www.gemplus.com/app/index.htm>> adresli internet sitesi (06 Eylül 1998)

EKLER

Ek. A GemClub Akıllı Kartı

GemClub, gelişmiş fonksiyonlara sahip kolaylıkla kullanılabilen mikroişlemcili bir karttır. Özel elektronik cüzdan, ölçümleme (kullanılan gaz veya elektriğin ölçülmesi), müşteri tanıma, vb gibi alanlardaki uygulamalarda kullanılmaktadır.

GemClub şunları içerir :

- 1 KB EEPROM içeren GemClub 1K kartları,
- EMV uyumlu GemClub-EMV kartları.

Veri Yapısı

GemClub, düz dosya yapısına sahiptir. Uygulama için gereken tüm dosyalar, bellekte aynı seviyede yer alır.

- Sistem dosyası: Kartın genel güvenlik yönetimi için kullanılır.
- Loyalty dosyaları: Puan ve kuralların saklanması için kullanılır.
- Kayıt Dosyaları: Veri depolama da kullanılır.
- Güvenlik dosyaları: Gizli kod (secret code) ve gizli anahtarların (secret key) depolanmasında kullanılır.

Komut Kümesi

GemClub işletim sistemi şu komutları içerir:

- Yönetim komutları: Nesne Yarat (Create Object), Nesne Yoket (Delete Object) ve Haberleşme Hızı Seç (Select Communication Speed)
- Uygulama Komutları: Ödül (Award), Geri Al (Redeem), Kuralı Kullan (Use Rule), Parametre Güncelle (Update Parameter) ve Parametre Oku (Read Parameter).

Veri Erişimi Yönetimi ve Güvenlik

Erişim koşulları veri dosyaları ve veri elemanlarını korumak için kullanılırlar. Bunlar, dosya için belirlenen koruma seviyesini de tanımlarlar. Genel olarak GemClub kartlarda yer alan dosyalar gizli kod veya gizli anahtar ile korunmaktadır.

Herbir veri dosyası veya veri elemanı için özel erişim koşulları belirlenebileceği gibi her komut veya komut grubu için de bu tanımlama yapılabilir.

Tek bir erişim koşulu bir byte üzerinde tanımlanır.

Gizli kodlar 8 byte uzunluğundadır ve 'Gizli Kod Dosyaları'nda saklanırlar. Her kod bir dosyada saklanır.

Başlıca erişim koşullarına (gizli kod, gizli anahtar) ek olarak PIN koruması da istenebilir. PIN, kart sahibine özel gizli bir koddur ve sistem dosyasından işaret edilen bir dosyada tutulur.

Gizli anahtarlara dayanan güvenli mesajlaşma mekanizması, kart ve terminal arasında gidip gelen verilerin doğruluğu ve tutarlılığını sağlamakta kullanılır. Üçlü DES (3DES - Triple DES) algoritmasını kullanır. Terminal ve kartın karşılıklı olarak birbirlerini doğrulamalarında da kullanılır.

Terminal, yolladığı komutla birlikte bunun güvenliğini sağlayacak bir de Mesaj Doğrulama Kodu (MAC_IN - Message Authentication Code_IN) gönderir. Kart, önce gelen MAC_IN'i yorumlar. Eğer doğru ise terminalin bu işlemi gerçekleştirmeye yetkisi vardır ve gelen veri tutarlıdır. Kart, gelen komut ile istenen işlevi yerine getirdikten sonra terminale MAC_OUT gönderir. Terminal MAC_OUT'a bakarak kartın doğruluğunu ve veri iletiminin tutarlılığını sınar.

Güvenlik için, gizli anahtar güncellenirken, yeni değer karta yollanmadan önce şifrelenir.

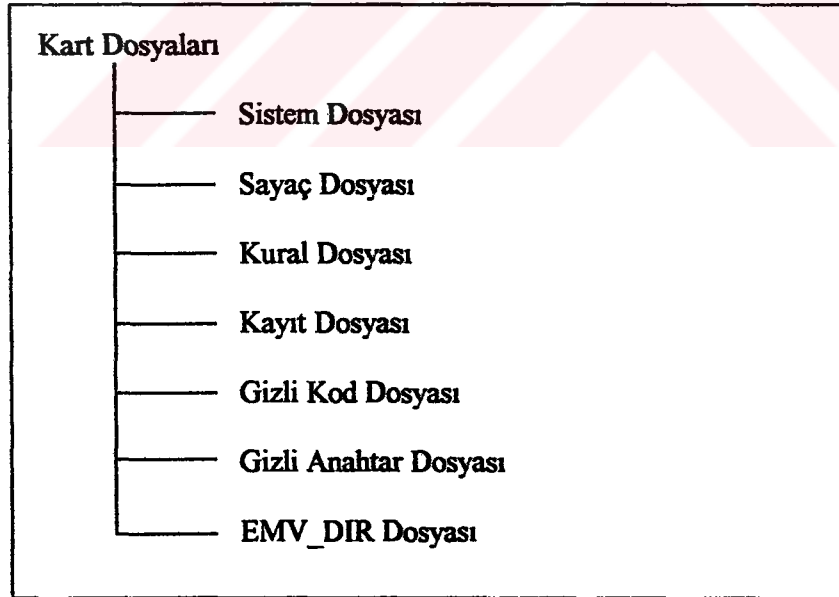
Uygulama komutundan sonra kartta işlemin sınaıma değeri (checksum) oluşturulur. Burada gizli anahtarlar ve 3DES algoritması kullanılır. Oluşan sınaıma değeri operatör tarafından işlemin doğrulanması için kullanılır.

Haberleşme

GemClub kartları ISO 7816-3'e uygun şekilde T=0 protokolüne göre veri gönderip alır. Ayrıca 9600 veya 115200 baud'luk yüksek hızlı iletişimi de destekler.

1. GemClub Dosyaları

Bu bölümde GemClub dosya yapısı ele alınacaktır. GemClub kartlarda düz dosya yapısı kullanılmaktadır. Bir başka deyişle, Şekil A.1'de de gösterilen tüm kart dosyaları aynı hiyerarşik seviyede yer almaktadır.



Şekil A.1 Dosya Yapısı

GemClub Dosya Organizasyonu

GemClub kartlarda aşağıdaki dosya tipleri desteklenmektedir:

- **Sistem Dosyası:** Her kartta bir tane sistem dosyası bulunmaktadır. Bu dosya kartın genel güvenlik yönetimi için kullanılmaktadır. Aynı zamanda PIN dosyasına işaretçi de burada saklanmaktadır.
- **İşlem Dosyası:** Sayaç dosyalarında puan, müşteri profili gibi bilgiler ile kural dosyalarında kuralların saklanması için kullanılırlar.
- **Güvenlik Dosyaları:** Gizli anahtar dosyaları ve gizli kod dosyaları.
- **Kayıt Dosyaları:** Veri depolanmasında kullanılırlar. ISO/IEC 7816-4 kayıt yönetimi standartlarına uygundur. Dosyaların adlandırılması ve dosyalara erişim bu standartlar çerçevesinde gerçekleştirilir.

Aşağıdaki dosya seçim yöntemleri kullanılmalıdır:

- **Dosya Tipi:** Veri dosyasının amacını belirler.
- **Kısa Dosya Belirteci (SFI – Short File Identifier):** 01h – 1Eh arasında değerler alan 5 bitlik bu sayı, dosya yaratıldığında oluşur. Sıfır değeri GemClub kartlarında geçerli değildir, reddedilecektir.

GemPlus Dosya Tipleri

GemClub işletim sisteminde Tablo A.1’de verilen dosya tipleri desteklenmektedir:

Tablo A.1 Dosya Tipleri

Dosya Tipi	Veri Nesnesi Tipi	Açıklama
Sistem Dosyası	01h	İşletim sistemi tarafından haricen yönetilen veri elemanları.
Kayıt Dosyası	02h	Terminal tarafından yönetilen kayıtlar.
Sayaç Dosyası	03h	Kart sahibinin hak ettiği puanlar.
Kural Dosyası	04h	Sayaçlar üzerinde yürütülebilecek makro-komutlar.
Gizli kod Dosyası	05h	Gizli kod.
Gizli Anahtar Dosyası	06h	Şifreleme algoritmasında kullanılan gizli değer.

- Her kartta sadece bir sistem dosyası olabilir.
- Sistem dosyasının SFI değeri her zaman 01h’dir.

- Aynı tipteki iki dosya aynı SFI değerine sahip olamaz. GemClub kartlarında en fazla 16 tane kural dosyası yaratılabilir. Sistem dosyası ve kural dosyası dışındaki tiplerden en fazla 30 dosya yaratılabilir.

Kişiselleştirme Bilgileri

Kişiselleştirme bilgileri dosyalarda tutulur ve dosyanın kendi yönetimi için kullanılır. Şu bileşenleri içerebilir:

- Erişim koşulları
- PIN kodu ve EMV katalog dosyası referansları
- İşlem sınaması için anahtar değer
- Kural versiyonu
- İzin verilen maksimum deneme sayısı ve gizli kod ile gizli anahtarların doğrulanması için kalan deneme sayısını içeren onay sayaçları.

Parametre Güncelleme komutu ile, bu bilgiler dosya yaratıldıktan sonra güncellenmelidir. Komut kullanılırken, güncellenecek veri verilen bir etiket ile belirtilir. İşlemi basitleştirmek için, tüm kişiselleştirme bilgilerini aynı anda güncellemek için kullanılacak 'kişiselleştirme etiketi' bulunmaktadır.

Bu bölümdeki dosya tanımlarında, etiket değerlerinin yer aldığı tablolarda kişiselleştirme veri elemanları gri olarak gösterilmektedir.

Sistem Dosyası

Sistem dosyasında, kart işlem sayacı (CTC – Card Transaction Counter), işletim sistemi ve kişiselleştirme bilgilerinin de dahil olduğu 12 byte'lık bilgi bulunabilir.

GemClub kartlarda sistem dosyalarının önceden tanımlanmış olmasına rağmen, kartın işlemleri gerçekleştirebilmesi için sistem dosyasının özelliklerinin belirtilmesi gerekmektedir.

Sistem dosyası yaratıldığında kartta başka bir dosya bulunmamaktadır. Bu dosyalar daha sonraki aşamalarda oluşturulur.

Sistem dosyasına ait bilgiler şöyledir:

Dosya Tanımlayıcısı	01h
Dosya Tipi (Veri nesnesi tipi)	Sistem (01h)
Dosya Büyüklüğü	12 byte

Sistem dosyasının detayları Tablo A.2'de verilmiştir.

Tablo A.2 Sistem Dosyası Yapısı

Etiket	Uzunluk	İçerik	Açıklama
21h	1 byte	Güncelleme / Yoketme için erişim koşulu	Sistem dosyasını güncellemek / yoketmek için gerekli yetkiyi belirler.
22h	1 byte	Okuma için erişim koşulu	Sistem dosyasını okumak için gerekli yetkiyi belirler.
23h	1 byte	Yaratma için erişim koşulu	Dosya yaratmak için gerekli yetkiyi belirler.
24h	1 byte	PIN kodu dosyası referans bilgisi	Kart sahibini tanımak amacıyla kullanılan PIN olarak kullanılan gizli kod dosyası için işaretçi.
26h	1 byte	EMV_DIR dosyası referans bilgisi	EMV_DIR simülasyonu için kullanılan dosya için işaretçi.
27h	2 byte	Kart işlem sayacı	Doğrulama işlemlerinde güvenli mesajlaşma için kullanılır.

Etiket, 1 byte uzunluğunda sistemde tek (unique) olan bir sayıdır. Kullanılacak verinin belirlenmesinde kullanılır ancak kartta saklanmaz.

Kişiselleştirme etiketi 20h'dir. Bu değer kullanılarak 21h, 22h, 23h, 24h ve 26h etiketli alanların eş zamanlı olarak okunması veya güncellenmesi yapılabilir.

İşlem Dosyaları

İşlem dosyaları, sayaç dosyaları ve kural dosyalarına verilen genel isimdir.

Sayaç Dosyaları

Sayaç dosyaları, ödül ve/veya redeem uygulamalarında yer alan puanların saklanması için kullanılırlar. Sayaç dosyasına erişim kural dosyası üzerinden geçilerek gerçekleştirilir. Tek bir GemClub kartında 30 tane değişik sayaç dosyası yaratılabilir.

GemClub kartlarındaki sayaç dosyaları, belirtilen özelliklere bağlı olarak farklı yapılarla sahip olabilirler. Tablo A.3'te verilen yapı alanı kullanılarak istenen tipte sayaç yaratılabilir.

Tablo A.3 Sayaç Dosyası Yapı Alanı

Bit 0	Toplam Bakiye
Bit 1	Ziyaret Sayacı
Bit 2	Geçerlilik Tarihi – ödül
Bit 3	Geçerlilik Tarihi – Redeem
Bit 4	Bu Sayaç İçin Belirtilen Kurallar
Bit 5	Başlık
Bit 6	(Gelecekteki Uygulamalar İçin)
Bit 7	(Gelecekteki Uygulamalar İçin)

Sayaç yapısına ekleme yapmak için ilgili bite ‘1’ değeri atanır. Bit ‘0’ değerinde ise ilgili veri nesnesi tipi sayaç yapısına dahil değildir.

Sayaç dosyasına ait bilgiler şöyledir:

Dosya Tanımlayıcısı 01h – 1Eh

Dosya Tipi (Veri nesnesi tipi) Sayaç (03h)

Dosya Büyüklüğü Belirtilen özelliklere göre 16-40 byte arasındır.

Sayaç dosyasında en azından Tablo A.4’te belirtilen detaylar yer almalıdır:

Tablo A.4 Sayaç Dosyası Yapısı

Etiket	Uzunluk	İçerik	Açıklama
61h	1 byte	Güncelleme / Yoketme için erişim koşulu	Sayaç dosyasını güncellemek / yoketmek için gerekli yetkiyi belirler.
62h	1 byte	Okuma için erişim koşulu	Sayaç dosyasını okumak için gerekli yetkiyi belirler.
63h	1 byte	İşlem Sınama için anahtar referansı	İşlem sınama değeri hesaplamasında kullanılacak anahtar dosyasına ait SFI. İşlem sınamasının istendiğini belirtir.
64h	1 byte	Ödül için erişim koşulu	Ödül puanı verme yetkisini belirtir.
65h	1 byte	Geri alma için erişim koşulu	Puanları geri alma yetkisini belirtir.
70h	3 byte	Bakiye	Sayaçta o an bulunan puanları gösterir.

Kişiselleştirme etiketi olan 60h kullanılarak 61h, 62h, 63h, 64h ve 65h etiketi ile gösterilen dosya detayları eş zamanlı olarak okunup güncellenebilir.

GemClub kartlarındaki sayaçlara eklenebilecek seçimlik özellikler Tablo A.5'te verilmiştir. Bu özellikler sayaç dosya yapısına uygun bir şekilde belirlenmelidir.

Tablo A.5 Sayaç Dosyası Seçimlik Özellikler

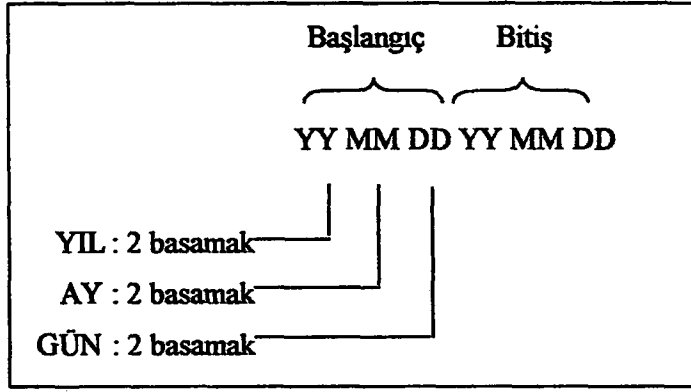
Etiket	Boyut	İçerik	Açıklama
71h	3 byte	Toplam Bakiye	Yaratılmasından bu yana bu sayaca verilen puanları gösterir.
72h	2byte	Ziyaret Sayacı	Yaratılmasından bu yana yürütülen ödül işlemleri sayısını verir.
73h	6byte	Ödül Geçerlilik Tarihi	Ödül işlemi için belirlenen geçerlilik tarihini verir. (Örneğin belirli bir tarihe kadar ödül veriliyorsa kullanılabilir)
74h	6 byte	Geri Alma Geçerlilik Tarihi	Geri alma işlemi için belirlenen geçerlilik tarihini verir. (Örneğin belirli bir tarihe kadar indirim uygulanıyorsa kullanılabilir)
75h	2byte	Bu Sayaç İçin İzin Verilen Kurallar	Bu sayaç üzerinde yürütülebilecek kuralları belirler. Bu alan boşsa ' <i>Kural Kullan</i> ' komutu ile sayaç detayları değiştirilemez. Aksi alde her bit bir kurala karşı gelir.
76h	8byte	Başlık	Sayacın adının ve sürümünün belirtildiği alfanümerik değer. Kart ömründe tekrar tekrar kullanılabilir.

Eğer sistem tarihi, belirlenmiş ödül/geri alma tarih aralığında değilse bu işlem reddedilir. Tarih bilgilerinin kodlanması aşağıda belirtilen kriter kontrol edilebilecek şekilde yapılmalıdır.

Başlangıç Tarihi <= Sistem Tarihi <= Bitiş Tarihi

İşletim sistemi, tarih formatları üzerinde hiçbir kontrol uygulamaz.

2000 yılı probleminin giderilmesi için GemClub kartlarında tarihler BCD formatında Şekil A.2'deki gibi tutulmaktadır.



Şekil A.2 GemClub Tarih Gösterimi

2000 yılının 1900 yılı olarak yorumlanmaması için terminal tarihi göndermeden önce Tablo A.6’da gösterildiği gibi değiştirir:

Tablo A.6 GemClub Tarih Dönüşümü

Yıl	Dönüşüm Sırası
1997	'00'
1998	'01'
1999	'02'
2000	'03'
2001	'04'

Kural Dosyaları

Kural dosyası, '*Kural Kullan*' komutu sırasında yürütülen makrokomut kümesidir. Kurallar kullanılarak en fazla dört sayaç birleştirilebilir. Bir GemClub kartında maksimum 16 tane kural dosyası olabilir.

Kural dosyasının yapısı, dosya için belirlenen özelliklere bağlıdır. Kural dosyalarına ait bilgiler şu şekildedir:

Dosya Tanımlayıcısı	01h – 10h
Dosya Tipi (Veri nesnesi tipi)	Kural (04h)
Dosya Büyüklüğü	İşletim sistemi bilgileri dahil 12 + 8n byte.

(n : makrokomut sayısı)

Kural dosyasında Tablo A.7’deki detaylar mutlaka yer almalıdır.

Tablo A.7 Kural Dosyası Yapısı

Etiket	Boyutluk	İçerik	Açıklama
81h	1 byte	Güncelleme / Yoketme için erişim koşulu	Kural dosyasını güncellemek / yoketmek için gerekli yetkiyi belirler.
82h	1 byte	Okuma için erişim koşulu	Kural dosyasını okumak için gerekli yetkiyi belirler.
83h	1 byte	İşlem Sınama için anahtar referansı	İşlem sınama değeri hesaplamasında kullanılacak anahtar dosyasına ait SFI. İşlem sınamasının istendiğini belirtir.
84h	1 byte	Kural kullanma için erişim koşulu	'Kural Kullan' komutu ile bu kuralı çalıştırma yetkisini belirler.
85h	1 byte	Sürüm	Kuralın sürümünü gösterir.
90h	8 byte	Makrokomut 1	Belirlenen sayaç zerinde yürütülecek hareketi belirtir. Sayaç için yaratılan her kural için en az bir makro yazılmalıdır.

Kişiselleştirme etiketi 80h'dir. Bu etiket kullanılarak 81h, 82h, 83h, 84h ve 85h etiketli detaylar aynı anda güncellenebilir ya da okunabilir.

Yaratılan diğer makrokomutlar da Tablo A.8'de tarif edilen şekilde dosyaya eklenebilir.

Tablo A.8 Kural Dosyasına Başka Makrokomut Eklenmesi

Etiket	Boyutluk	İçerik	Açıklama
91h	1 byte	Makrokomut 2	Bu sayaç için kullanılacak komutu tanımlar.
92h	1 byte	Makrokomut 3	Bu sayaç için kullanılacak komutu tanımlar.
93h	1 byte	Makrokomut 4	Bu sayaç için kullanılacak komutu tanımlar.

Kural dosyasında en fazla dört makrokomut saklanabilir.

Belirli bir sayaç için tanımlanan işlemin (ödül veya indirim) mantıksal ifadesine 'Makrokomut' adı verilir. Makrokomutlar, Tablo A.9'da verilen bilgilerden oluşurlar.

Tablo A.9 Makrokomut Yapısı

Alan	Boyutluk	Açıklama
Fonksiyon	1 byte	Belirlenen sayaca uygulanacak işlem. 00h – Makrokomutu geç 01h – Komutun ödül işlemi için olduğunu belirtir. 02h – Komutun indirim işlemi için olduğunu belirtir. (Diğer değerler gelecekteki kullanımlar için ayrılmıştır.)
Sayaç Referansı	1 byte	Makrokomutun üzerinde yürütüleceği sayaç dosyasının SFI değeri.
Parametre A	3 byte	İlk dönüşüm parametresi. Parametre B’de belirtilen değere erişildiğinde sayaca eklenecek veya sayaçtan çıkarılacak puan sayısını verir.
Parametre B	3 byte	İkinci dönüşüm parametresi. Kart için belirtilen sayaça puan eklemekten ya da çıkarmadan önce erişilmesi gereken değeri verir.

Makrokomutun yürütülebilmesi için sayaçtaki ‘izin verilen kurallar’ alanında bu kurala izin verilmiş olması gerekmektedir.

Güvenlik Dosyaları

Güvenlik dosyaları, gizli kod ve gizli anahtar dosyalarından oluşmaktadır.

Gizli Kod Dosyaları

Gizli kodlar dosya korumada kullanılırlar. Herbir gizli kod dosyasında sadece bir gizli kod bulunabilir. Bunların yanısıra kullanılabilecek olan PIN kodu da bir gizli koddur ve hangi dosyada bulunduğu sistem dosyasında belirtilmektedir. Bir dosyanın gizli kod veya PIN kodu ile korunabilmesi için o dosyanın erişim koşulunda söz konusu gizli kod veya PIN koduna yapılan bir referans yer almalıdır.

Gizli kod dosyalarına ait bilgiler şu şekildedir:

Dosya Tanımlayıcısı	01h – 1Eh
Dosya Tipi (Veri nesnesi tipi)	Gizli Kod (05h)
Dosya Büyüklüğü	İşletim sistemi bilgileri dahil 16 byte

Gizli kod dosyasında Tablo A.10’da verilen alanlar bulunmalıdır.

Tablo A.10 Gizli Kod Dosyasının Yapısı

Etiket	Boyutluk	İçerik	Açıklama
A1h	1 byte	Güncelleme / Yoketme için erişim koşulu	Gizli kod dosyasını güncellemek / yoketmek için gerekli yetkiyi belirler.
A3h	1 byte	Maksimum deneme sayısı ve Kalan deneme sayısı	İzin verilen maksimum deneme sayısı ve kart reddedilip gizli kodun engellenmesine kadar kalan deneme hakkı sayısı.
B0h	8 byte	Gizli kod değeri	Gizli kod değeri.

Kişiselleştirme etiketi A0h'dir. Bu etiket kullanılarak A1h ve A3h etiketli detaylar aynı anda güncellenebilir.

Maksimum deneme sayısı olarak en fazla 15 verilebilir. Kalan deneme sayısı da en fazla 15 olabilir. Gizli kodun doğru olarak verilmesiyle birlikte kalan deneme sayısı, maksimum deneme sayısına eşitlenir.

Gizli Anahtar Dosyaları

GemClub'daki gizli anahtar değerleri güvenli mesajlaşmanın sağlanması ve işlem sınama değerinin hesaplanmasında kullanılırlar. Güvenli mesajlaşma amaçlı gizli anahtar kullanımı için ilgili SFI değerinin korunacak dosyanın erişim koşulunda belirtilmesi gerekmektedir. Erişim koşulunda da güvenli mesajlaşmanın kullanılacağı belirtilmelidir. Gizli anahtar işlem sınama değeri hesabında kullanmak için SFI, ilgili kural veya sayaç dosyasında belirtilmelidir. Herbir gizli anahtar dosyasında sadece bir anahtar değeri yer alabilir.

Gizli anahtar dosyalarına ait bilgiler şu şekildedir:

Dosya Tanımlayıcısı	01h – 1Eh
Dosya Tipi (Veri nesnesi tipi)	Gizli Anahtar (06h)
Dosya Büyüklüğü	İşletim sistemi bilgileri dahil 24 byte

Gizli anahtar dosyasında Tablo A.11'deki alanlar bulunmalıdır:

Tablo A.11 Gizli Anahtar Dosyasının Yapısı

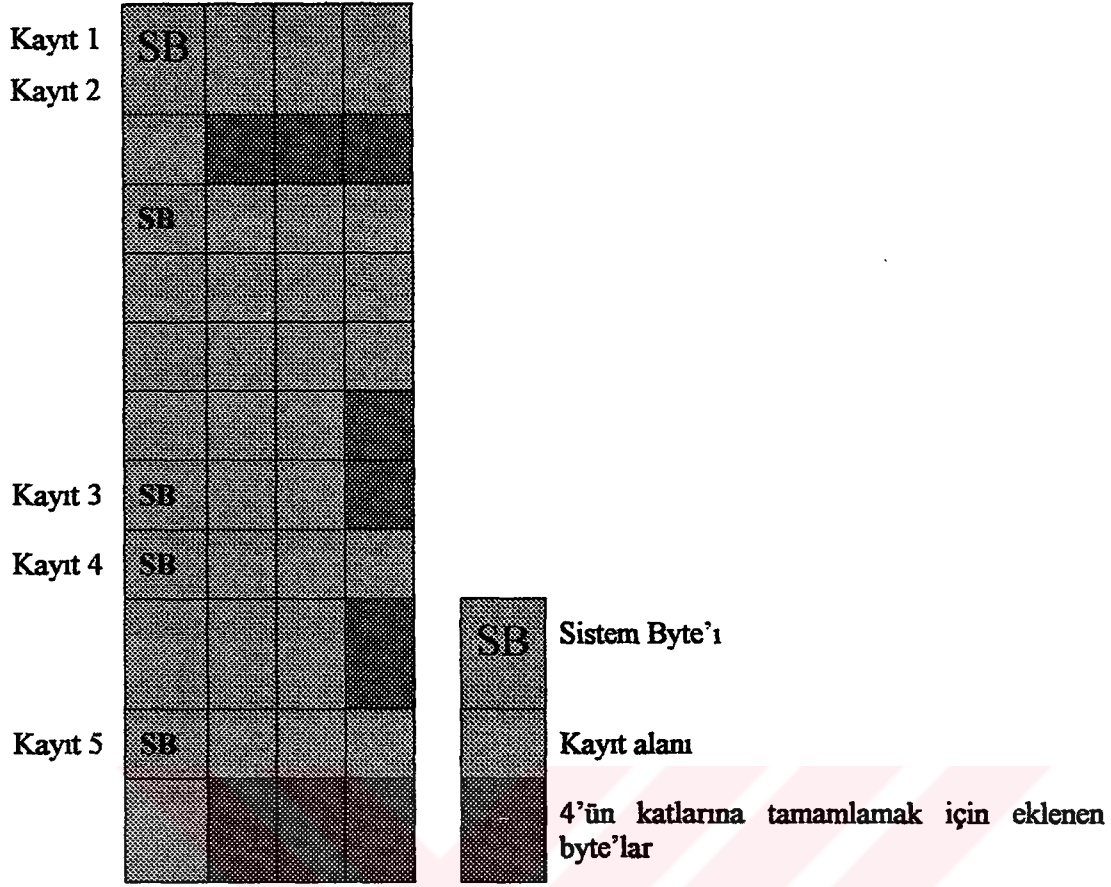
Etiket	Uzunluk	İşlev	Açıklama
C1h	1 byte	Güncelleme / Yoketme için erişim koşulu	Gizli anahtar dosyasını güncellemek / yoketmek için gerekli yetkiyi belirler.
C3h	1 byte	Maksimum deneme sayısı ve Kalan deneme sayısı	İzin verilen maksimum deneme sayısı ve kart reddedilip gizli anahtarın engellenmesine kadar kalan deneme hakkı sayısı.
D0h	8 byte	Gizli anahtar değeri	3DES standardına uygun olarak,3DES anahtarı olarak kullanılacak değerin yarısı. Güncelleme işlemi sırasında gizli anahtarın iki yarısına ayrı ayrı erişilir.
D1h	8 byte	Gizli anahtar değeri	3DES standardına uygun olarak,3DES anahtarı olarak kullanılacak değerin yarısı. Güncelleme işlemi sırasında gizli anahtarın iki yarısına ayrı ayrı erişilir.

Kişiselleştirme etiketi C0h'dir. Bu etiket kullanılarak C1h ve C3h etiketli detaylar aynı anda güncellenebilir.

Maksimum deneme sayısı olarak en fazla 15 verilebilir. Kalan deneme sayısı da en fazla 15 olabilir. Gizli anahtarın doğru olarak verilmesiyle birlikte kalan deneme sayısı, maksimum deneme sayısına eşitlenir.

Kayıt Dosyaları

Kayıt dosyaları, GemClub kartlarında verilerin depolandığı yerlerdir. Kartlarda Şekil A.3'te de gösterildiği gibi değişken uzunluklu kayıtlar içeren doğrusal dosyalar kullanılmaktadır.



Şekil A.3 Değişken uzunluklu kayıtlar içeren doğrusal dosya

Dosyadaki ilk kayıt '1' olarak numaralandırılmıştır. Doğrusal değişken kayıt dosyasında en fazla 254 kayıt bulunabilir. Bir kayıt en fazla 255 byte uzunlukta olabilir. Güvenli mesajlaşma durumunda kaydın maksimum uzunluğu 24 byte ile kısıtlıdır.

Kayıt dosyasının yapısı belirlenen özelliklere göre değişir. Kayıt dosyasına ait bilgiler şöyledir:

Dosya Tanımlayıcısı	01h – 1Eh
Dosya Tipi (Veri nesnesi tipi)	Kayıt (02h)
Dosya Büyüklüğü	(Değişken)

Kayıt dosyasında Tablo A.12'de verilen alanlar bulunmalıdır.

Tablo A.12 Kayıt Dosyasının Yapısı

Etiket	Uzunluk	İçerik	Açıklama
41h	1 byte	Güncelleme / Yoketme için erişim koşulu	Kayıt dosyasını güncellemek / yoketmek için gerekli yetkiyi belirler.
42h	1 byte	Okuma için erişim koşulu	Kayıt dosyasını okumak için gerekli yetkiyi belirler.
	L ₁	Kayıt 1	Veri
	L ₂	Kayıt 2	Veri
	L _n	Kayıt N	Veri

Kişiselleştirme etiketi 40h'dir. Bu etiket sayesinde 41h ve 42h etiketlerine sahip alanlar aynı anda güncellenebilir, okunabilir.

Kayıt dosyasının uzunluğu, dosya tanımlayıcısı ve erişim koşulları uzunluğu ile kayıtların uzunluğunun toplamıdır. Kayıt dosyasında yer alan kaydın uzunluğu ise şöyle ifade edilebilir:

Kayıt uzunluğu = (1 byte - SB) + (veri için kullanılan byte'lar) + (4'e tamamlama için kullanılan byte'lar)

Bu hesaplama göre Şekil A.3'te gösterilen dosyanın uzunluğu =

8 byte (dosya tanımlayıcısı) +

8 byte (Kayıt 1 : 1 byte SB+4 byte kullanılan+3 byte tamamlama) +

16 byte (Kayıt 2 : 1 byte SB+14 byte kullanılan+1 byte tamamlama) +

4 byte (Kayıt 3 : 1 byte SB+2 byte kullanılan+1 byte tamamlama) +

8 byte (Kayıt 4 : 1 byte SB+6 byte kullanılan+1 byte tamamlama) +

8 byte (Kayıt 5 : 1 byte SB+4 byte kullanılan+3 byte tamamlama) = 52

byte'tır.

İsimle Referans Verme (EMV Simülasyonu)

EMV özellikleri sadece GemClub-EMV kartlarıyla kullanılabilir. EMV kart spesifikasyonlarında belirtilen '*Uygulama dosyasını ismiyle seç*' komutunun simülasyonu gerçekleştirilerek GemClub-EMV kartlarının bu standarda uyumu sağlanmıştır.

ISO/IEC 7816-4 standardı ile belirtilen isimler ile katalog seçimi de GemClub kartları tarafından desteklenmektedir. GemClub tarafında kabul edilen kayıt dosyası isimleri EMV_DIR dosyasında da kayıtlı olmalıdır.

Uygulama dosyasının adı, isimle seçim için sağdan kesilebilir.

Ödeme Sistemi Ortamı (PSE – Payment System Environment) de isimle seçim için sağdan kesilebilir.

Tarama algoritması karttaki EMV_DIR dosyasının içeriğini tarar. Algoritma, uyumlu ilk kaydı bulduğunda durur.

GemClub kartları, EMV standartlarına göre yapılandırılmış dosyaları da destekler. EMV_DIR uyumlu kayıt dosyasının yaratılmasına örnek aşağıda verilmiştir.

EMV_DIR dosyası, sistem dosyasında belirlenen ve EMV simülasyonunda görev alan bir kayıt dosyasıdır. Dosyadaki ilk kayıt EMV PSE yapısına (1PAY.SYS.DDF01) uygun olmalıdır. Etiket olarak 9Dh kullanılmalıdır.

EMV_DIR dosyasındaki her kayıt, EMV kart spesifikasyonları uyarınca basit bir etiket boyu değeri (TLV – Tag Length Value) kullanılarak yapılandırılmalıdır. EMV_DIR dosyasındaki zorunlu alanların başlangıcı Tablo A.13'te gösterilmiştir:

Tablo A.13 EMV_DIR Dosyasının Yapısı

	Etiket	Uzunluk	İçerik	Açıklama
Kayıt 1	70h	1 byte	Veri Taslağı	
= DDF	Değişken	1 byte	Uzunluk	Veri Taslağı
	61h	1 byte	Uygulama Taslağı	
	Değişken	1 byte	Uzunluk	Uygulama Taslağı
	9Dh	1 byte	Etiket	Katalog Tanım Dosyası Adı (DDF – Directory Definition File) Etiket = 9Dh
	5-16	1 byte	Uzunluk	Uygulama Tanımlayıcısı
	-	5-16 byte	Uygulama Tanımlayıcısı	Kayıt dosyası tarafından tanınan uygulama adı.
	50h	1 byte	Etiket	Uygulama Başlığı
	1-16	1 byte	Uzunluk	Uygulama Başlığı
	-	1-16 byte	Uygulama Başlığı	Alfanümerik değer.

Kayıt 2	70h	1 byte	Veri Taslağı	
= ADF	Değişken	1 byte	Uzunluk	Veri Taslağı
	61h	1 byte	Uygulama Taslağı	
	Değişken	1 byte	Uzunluk	Uygulama Taslağı
	4Fh	1 byte	Etiket	Uygulama Tanım Dosyası Adı (ADF – Application Definition File) Etiket = 4Fh
	5-16	1 byte	Uzunluk	Uygulama Tanımlayıcısı

-	5-16 byte	Uygulama Tanımlayıcısı	Kayıt dosyası tarafından tanınan uygulama adı.
50h	1 byte	Etiket	Uygulama Başlığı
1-16	1 byte	Uzunluk	Uygulama Başlığı
-	1-16 byte	Uygulama Başlığı	Alfanümerik değer.

Kayıt 3 = ADF	...	...	...	
------------------	-----	-----	-----	--

2. GemClub Başlangıç Durumu

Bu bölümde GemClub kartlardaki tüm bileşenlerin başlangıç durumları ele alınmıştır.

Koşullama İşlemi

İmalat sırasında, mikroişlemcili kartların koşulanmaları gerekmektedir. Bu aşamada karta aşağıda belirtilen temel bilgiler yazılır:

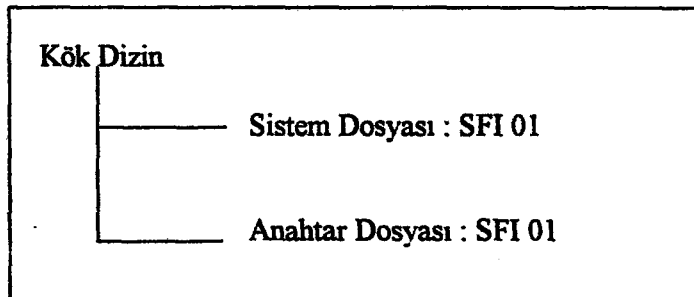
- Kart seri numarası ve Kart tedarikçisi referansı
- Sistem dosyası yapısı
- Koruma anahtarları
- Filtre yaratılması (GemClub-EMV kartları için)

İki çeşit kart koşullama işlemi bulunmaktadır:

- Müşteri kartı için: Kartlar farklı anahtarlar tarafından korunmaktadır.
- Örnek kartlar için: Geliştirme setleriyle birlikte gönderilen kartlar ortak bir anahtar ile korunmaktadır.

Başlangıçtaki Dosya Yapısı

Şekil A.4'te başlangıç durumunda GemClub kartı üzerindeki dosya yapısının nasıl olduğu gösterilmiştir.



Şekil A.4 Başlangıçtaki Dosya Yapısı

Sistem Dosyası

Karta ilişkin genel erişim koşulları sistem dosyasında tanımlanmaktadır. Dosya parametreleri Tablo A.14'te verilmiştir.

Tablo A.14 Sistem Dosyası Başlangıç Parametreleri

Hareket Erişim Koşulları ve Parametreleri	Çalışık Kart ve Müşteri Kartı Prosedürleri
Güncelleme / Yoketme	Anahtar 01
Okuma	Serbest
Yaratma	Anahtar 01
PIN Kodu Dosyası Referansı	00h
EMV_DIR Dosyası Referansı	00h
Kart İşlem Sayacı	0005h

Anahtar Dosyası (SFI 01)

Bu dosyada, kart uygulaması yaratıcısı sağlayıcısı tarafından kişiselleştirme işlemlerinde kullanılan 01 anahtarı bulunmaktadır. Değeri, kartın başlangıç durumuna bağlıdır.

Örnek proses tarafından koşullama yapılmış kartlarda sabit ASCII katarı 'TESTKEY1TESTKEY2'dir.

Müşteri kart prosesi tarafından ilklendirilen uygulama kartlarındaki anahtar, müşteri kartında yer alan müşteri anahtarı kullanılarak oluşturulan farklı bir değere sahiptir.

Dosya parametreleri Tablo A.15'te verilmiştir.

Tablo A.15 Anahtar Dosyası Parametreleri

Değer	Uzunluk
01h	24 byte

Güvenlik erişim koşulları ise Tablo A.16'da gösterildiği gibidir.

Tablo A.16 Erişim Koşulları

İşlem	Erişim Koşulu
Güncelleme / Yoketme	Anahtar 01 ile Güvenli Mesajlaşma

Sayaçlardaki maksimum deneme ve kalan deneme sayılarına ilişkin parametreler '3' değerine ayarlanmıştır.

Kart Seri Numarası

Kart seri numarası her bir GemClub kartı için tektir ve değiştirilemez. Şekil A.5'te gösterildiği gibi 8 byte'lık alana 32 bitlik iki sözcük olarak kodlanmıştır.

Bitler																																																																																
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00																																																	
[1										10 bit										] [2													22 bit										]																																					
[3										10 bit										] [4													6 bit						] [5													8 bit								] [6													8 bit]							
1 : Batch numarası (10 bit)																																																																																
2 : Seri Numarası (22 bit)																																																																																
3 : Seri Numarası (10 bit)																																																																																
4 : Hafta Numarası (06 bit)																																																																																
5 : Seri Numarası (08 bit)																																																																																
6 : Üretim Yılı (08 bit)																																																																																

Şekil A.5 Seri Numarası

2, 3 ve 5. alanlar birleşerek seri numarasını oluştururlar. Seri numarası salt-okunabilir bir parametredir.

Kart Tedarikçisi Referans Numarası

Kart tedarikçisi referans numarası üç byte ve sınav kontrolü byte'ından müteşekkil olmak üzere toplam dört byte'tan oluşmaktadır. Bu sayı her müşteri için Gemplus tarafından verilmektedir; daha sonra değiştirilemez. Örnek kartlarda kart tedarikçisi referans numarası 00 FF 00 10h olarak belirlenmiştir.

G/Ç Tampon Uzunluğu

Güvenli mesajlaşma kullanılmadığı zaman bir seferde 255 byte'lık bir kaydı güncellemek mümkündür; ancak terminal veya okuyucu kaynaklı kısıtlamalar sonunda bu rakam daha düşük olabilir. Güvenli mesajlaşma kullanılıyorsa en fazla 24 byte'lık veri söz konusudur.

EEPROM Büyüklüğü

Tablo A.17'de belirtilen bellek büyüklükleri koşullama sonrası değerlerdir.

Tablo A.17 EEPROM Büyüklüğü

Ürün	Sistem ve anahtar dosyaları olmadan EEPROM	Sistem ve anahtar dosyaları oluştuktan sonra kalan EEPROM
GemClub 1K	952 byte	916 byte
GemClub EMV	748 byte	712 byte

3. GemClub Güvenlik Mimarisi

GemClub kartlarda bulunan güvenlik mekanizmaları beş başlıkta toplanabilir:

- Güvenlik durumları
- Güvenlik özellikleri: Kartta bulunan dosyalar üzerinde yapılan hareketlerin yetki kontrolünü sağlayan güvenlik spesifikasyonları.
- Güvenlik mekanizmaları: Karttaki dosyaların güvenliğini sağlayan prosedürler. Bunlar, ayrıca kart ve terminal arasındaki haberleşmenin bozuk olmamasını garantilerler.
- İşlem sınamaları: Operatöre, işlemlerin doğruluğunu bildiren prosedürler.
- Erişim koşulu kontrol algoritması: Kullanıcı veya kart sağlayıcının, kart üzerinde işlem yapmaya yetkili olup olmadığını gösteren algoritma.

Güvenlik Durumları

Komut yürütülmesi sırasında, güvenli mesajlaşmanın başarıyla tamamlanması gibi durumları kaydetmek gerekebilir. Bu kayıtlara 'Güvenlik Durumu' adı verilir. İki çeşit güvenlik durumu vardır:

- Komuta özel güvenlik durumu: Anahtar doğrulamaları kaydederler. Ayrıca, komutun veri alanlarında bulunması durumunda, gizli kod doğrulamalarını da saklarlar. Bu durum kaydı, komutun yürütülmesi sırasında RAM'de tutulur. Bunların, genel güvenlik durumu üzerinde bir etkisi yoktur.
- Genel güvenlik durumu: Komuttan önce yürütülen '*Gizli Kod Doğrula*' komutu sırasında, gizli kod verilirken kullanılır. PIN doğrulaması ve gizli kodun komut alanında iletilmediği ISO 7816-4 standartında tanımlanan komutlarda bu duruma ihtiyaç duyulmaktadır. Genel güvenlik durumunda, PIN kodu ve belirtilen son gizli kodun kaydı yer almaktadır. '*Gizli Kod Doğrula*' komutu kullanılmadan verilen gizli kodlar bu durumda yer almaz. RAM'de tutulan genel güvenlik durumu kart oturumunun sonunda silinir.

Güvenlik Özellikleri

GemClub kullanıcılarının, karttaki dosyalar üzerinde herhangi bir işlem yapmadan önce yerine getirmeleri gereken, önceden tanımlanmış güvenlik isterlerine 'Güvenlik Özellikleri' adı verilir. Kart üzerinde dosya yarattıktan sonra bunlar için geçerli olacak erişim koşulları belirlenebilir. Erişim koşulları, belirli bir fonksiyon veya fonksiyon grubu için tanımlanır. İlgili veriyi korumada kullanılan gizli kod ve gizli anahtarları kullanır.

Erişim koşulu tanımlanırken Tablo A.18'deki fonksiyonlar kullanılabilir:

Tablo A.18 Eriřim kořulları iin kullanılan fonksiyonlar

Fonksiyon	Aıklama	Veri Nesnesi
Yaratma	Veri objesi yaratma.	Hepsi
Güncelleme/Yoketme	Veri elemanı güncelleme veya veri nesnesi yoketme. Kayıt güncelleme/ekleme.	Hepsi
Okuma	Veri elemanı/kaydı okuma.	Güvenlik dosyaları ıřında hepsi.
Ödöl	Ödöl komutu yürütme.	Sayalar
İndirim	İndirim komutu yürütme.	Sayalar.
Kural Kullan	'Kural Kullan' komutu ürtme.	Kurallar.

Güncelleme ve yoketme fonksiyonları aynı erişim kořulu ile korunur.

Eriřim kořulu, bir byte üzerinde Şekil A.6'da gösterildiğı gibi tanımlanır:

b7	b6	b5	b4	b3	b2	b1	b0
PIN	Koşul		Kısa Dosya Tanımlayıcısı - SFI				
B7 : 1 – PIN kodu zorunlu 2 – PIN kodu zorunlu değil. B6-b5 : 00 – Gizli kod koruması ya da güvenli mesajlaşma gerekli değil. 01 – Gizli kod koruması gerekli 10 – Güvenli mesajlaşma gerekli 11 – Erişim kilitlenir. B4-b0 : b6-b5’te belirtilen koşulda geçen gizli kod veya gizli anahtar dosyasının tanımlayıcısı.							

Şekil A.6 Eriřim kořulu yapısı

PIN olarak kullanılacak gizli kod dosyası sistem dosyasında belirtilmektedir.

GemClub'da erişim kořulları iin belirlenen kodlara ait kurallar Tablo A.19'da verilmiştir.

Tablo A.19 Eriřim kořullarında yer alan kodlar için kurallar

Gerekenler	Yapılacak
Gizli kod koruması ya da güvenli mesajlaşma gerekmiyor.	x00xxxxxb kullanılmalıdır.
Belirli bir fonksiyon için 'Eriřim Aç'	000xxxxxb kullanılmalıdır.
Belirli bir fonksiyonun kilitli olduğunu kontrol et	x11xxxxxb kullanılmalıdır.
Sadece PIN kod kontrolü	100xxxxxb kullanılmalıdır.
Sadece güvenli mesajlaşma	010xxxxxb kullanılmalıdır.
Sadece gizli kod	001xxxxxb kullanılmalıdır.
Güvenli mesajlaşma ve PIN kontrolü	110xxxxxb kullanılmalıdır.
Gizli kod ve PIN kontrolü	101xxxxxb kullanılmalıdır.

Gizli kodlar, her erişim kořulu grubu için dosyalara erişimi güvenli kılarlar. Sistem dosyasında PIN kodu referansı olarak 00h belirtilmişse erişim kořulu doğrulaması sırasında PIN kodu kontrolü es geçilir. Okuma işlemleri için güvenli mesajlaşma desteklenmemektedir. Eğer okumalar için güvenli okuma seçeneğı belirtilirse bu 'kilitli' okuma olarak algılanır.

GemClub kartlarındaki her dosyanın erişim kořulları vardır. Her bir erişim kořulu dosyaya tanımlı komut/fonksiyon grubu için tanımlanır.

Bazı veri elemanlarına ilişkin okuma işlemi için işletim sisteminde bir erişim kořulu yer almaktadır.

Tablo A.20'de gösterilen veri elemanlarının okunması için işletim sisteminde dolaylı erişim kořulu bulunmaktadır.

Tablo A.20 Dolaylı Eriřim Kořuluna Sahip Veri Elemanları

Veri Nesnesi	Veri Elemanı	Okuma İçin Eriřim Kořulu
Gizli kod dosyası	Tüm veri elemanları	Kilitli
Gizli anahtar dosyası	Tüm veri elemanları	Kilitli

Güvenlik Mekanizmaları

Güvenlik mekanizmaları, karttaki dosyaların güvenliğini sağlamak için kullanılan yordamlardır. Bu mekanizmalar sayesinde kartın yetkisiz kullanımı da engellenmiş olur.

Güvenlik Mekanizmalarının Çalışma Şekli

Kart, üzerindeki ilgili gizli kod ile terminalden aldığı kodu karşılaştırır. İşletim sisteminin kullanacağı dahili gizli kod şunlardan biri olabilir:

- Komut içinde belirtilmiş (*'Gizli Kod Doğrula'* komutu ile)
- Kart tarafından dolaylı olarak bilinen (*'PIN Kodu Doğrula'* komutu ile)
- Komutta geçen veri nesneleri için belirtilen erişim koşulları içinde (*'Ödül'*, *'Kuralı Kullan'* komutları ile)

Gizli Kod İle Varlık Doğrulama

GemClub'da her gizli kod için tutulan 'Gizli Kod Onay Sayacı' kullanılır. Bu sayaç sayesinde gizli kod üzerinde yapılabilecek mükerrer saldırılar önlenabilir. Sayaç şu elemanlardan oluşur:

- İzin verilen azami deneme sayısı: Dosyaya erişim sırasında ardışıl olarak gerçekleşecek azami yanlış deneme sayısını verir. Bu parametrenin değeri en fazla 15 olarak belirtilebilir.
- Kalan deneme sayısı: Dosya erişiminin engellenmesine kaç deneme kaldığını gösterir. Parametre değeri en fazla 15 olabilir. Başarılı bir denemeden sonra bu sayı, izin verilen maksimum deneme sayısı ile güncellenir. Kalan deneme sayısı değeri, her zaman izin verilen değere eşit ya da bu değerden daha küçüktür.

Gizli Kod Yönetim Kuralları

Gizli kod yönetiminde aşağıdaki kurallar geçerlidir:

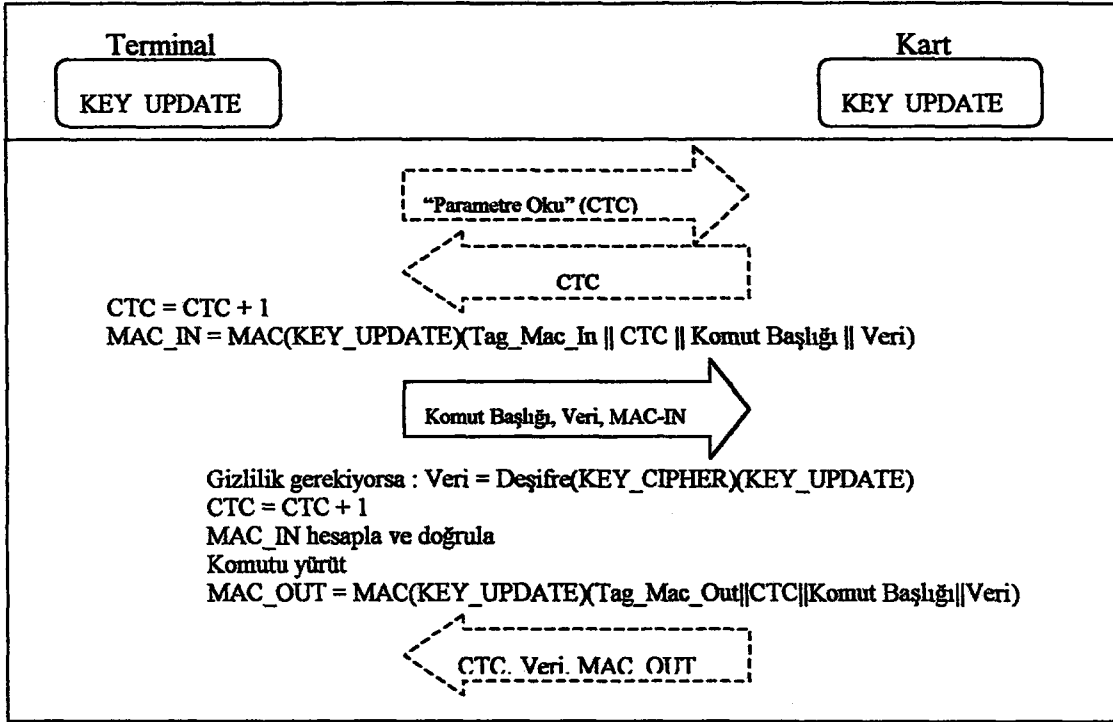
- Gizli kod bloke olmuşsa doğrulama yapılamaz, çünkü maksimum deneme sayısına ulaşılmıştır.
- Sunulan bir gizli kod ile karttaki gizli kod karşılaştırılmadan önce EEPROM'daki izin verilen deneme sayısı bir azaltılır.
- Yanlış gizli kod verilmesi nedeniyle gizli kodun bloke olması durumu dışında, yanlış bir gizli kod girişi genel güvenlik durumunu etkilemez. Bu duruma işletim sistemi genel güvenlik durumunda saklanan bilgileri iptal eder.
- Bloke olmuş bir gizli kodu etkinleştirmek için 'Kalan Deneme Sayısı' değerini güncellemek gerekir. Bunun için de güncelleme/yoketme erişim koşulunun sağlanması zorunludur. Bu da sadece gizli kod dosyası kendisi tarafından korunmuyorsa mümkündür.

Veri Doğrulama – Güvenli Mesajlaşma

Söz konusu mekanizma ile kart ile terminal arasındaki veri iletiminin doğruluğu ve tutarlılığı ve anahtar güncellemelerinin gizliliği sağlanmış olur.

Terminal, korunacak komut ile birlikte MAC_IN değerini de gönderir. Bu değeri alan kart da sınaama işlemlerini gerçekleştirecek, böylece terminalin ilgili işlemi yapmaya yetkili olup olmadığını ve verinin tutarlı bir biçimde iletilip iletilmediğini öğrenebilecektir.

'Parametre Güncelle' komutunun doğrulanmasına ilişkin bir örnek Şekil A.7'de yer almaktadır.



Şekil A.7 Parametre Güncelleme için güvenli mesajlaşma

Birinci Adım:

Terminal, ilgili işlem için önceden belirlenmiş olan gizli anahtar değerini kullanarak MAC_IN değerini hesaplar. MAC_IN değerinde aşağıdaki detaylar bulunmalıdır:

- Tag_Mac_In
- CTC
- Komut başlığı
- Karta gönderilecek mesaj

Anahtar değerinin güncellenmesi için gizlilik gerekiyorsa terminal mesajı DES3 algoritmasını kullanarak şifreler.

İkinci Adım:

İkinci adımda, kart (gizlilik gerekiyorsa) gelen mesajı deşifre eder; hesaplamalardan sonra terminalden aldığı MAC değerini doğrular.

Komutun sonunda, kart ilgili işlem için önceden tanımlanmış, MAC_IN hesabındaki değer olan gizli anahtarı kullanarak MAC_OUT'u hesaplar. Bu MAC_OUT değeri içinde şu detay bilgileri yer alır:

- Tag_Mac_Out
- CTC
- Komut Başlığı
- Terminale gönderilecek mesaj

Doğrulama işlemi başarı ile tamamlanırsa, terminalin ilgili işlemi yürütmeye yetkili olduğu anlaşılır. Bu durumda güvenli mesajlaşmaya ilişkin onay sayacı sıfırlanır. Doğrulamanın başarısız olması durumunda kart kalan deneme sayısını bir azaltır ve komutu reddeder.

Üçüncü Adım:

Terminal MAC_OUT değerini doğrular.

Gizli Anahtar Onay Sayacı

'Gizli anahtar onay sayacı' her gizli anahtar dosyasında bulunur ve bu dosya üzerinde gerçekleştirilecek mükerrer saldırıları önleneyi amaçlar. Sayaç şu bileşenleri içerir:

- İzin verilen maksimum deneme sayısı: Dosyaya erişimde izin verilen maksimum ardışıl yanlış deneme sayısıdır. Bu değer en fazla 15 olabilir.
- Dosyaya erişimin bloke edilmesine kaç deneme kaldığını gösterir. Başarılı bir doğrulama işleminden sonra, izin verilen maksimum deneme sayısına eşitlenir.

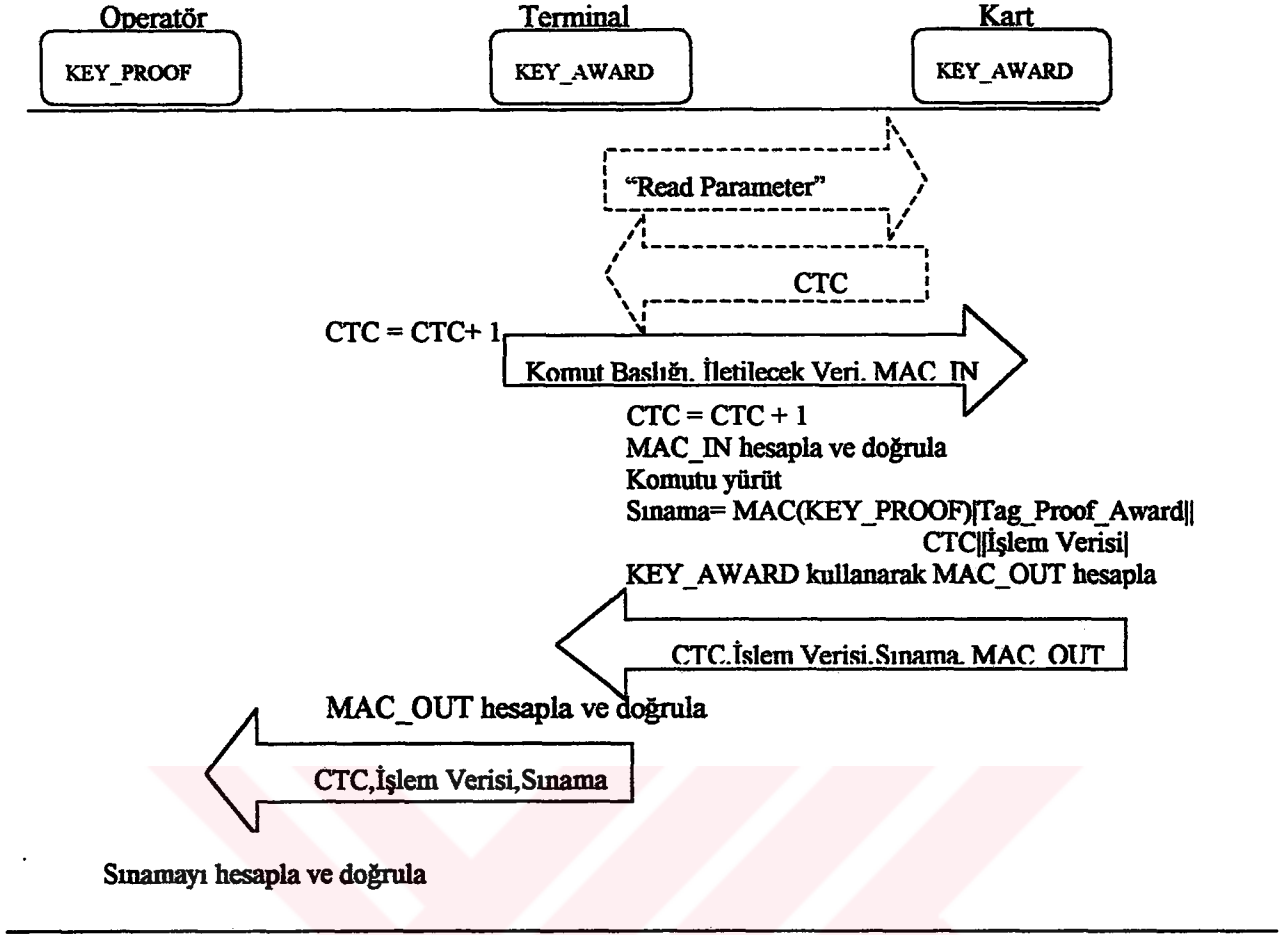
Kalan deneme sayısı her zaman izin verilen maksimum deneme sayısından küçüktür veya bu sayıya eşittir.

Gizli anahtar bloke olduğunda dosyaya erişim de engellenir. EEPROM'da tutulan sayaç değeri, terminalden gelen MAC_IN ve kartta hesaplanan MAC_IN değerlerinin karşılaştırılmasından önce bir azaltılır.

İşlem Sınama

GemClub kartları tarafından belirli işlemlerden sonra üretilen işlem sınama değeri, operatöre işlemin gerçekliğini gösterir. Örneğin, 'Ödül' işlemi sınamaya tabi tutulabilir.

Şekil A.8'deki örnekte kart, operatör tarafından sınanacak bir değer üretmektedir. Söz konusu işlem 'Ödül', 'Geri Alma' veya 'Kural Kullan' olabilir. Örnekte veri doğrulama adına güvenli mesajlaşma kullanılmaktadır. Ancak, veri doğrulama dışında da işlem sınama değeri üretilebilir.



Şekil A.8 – İşlem Sınama

Birinci Adım:

Terminal, ilgili işlem için önceden belirlenmiş olan gizli anahtar değerini (KEY_AWARD) kullanarak MAC_IN değerini hesaplar. MAC değerinde aşağıdaki alanlar bulunur:

- Tag_Mac_Award_In
- CTC
- Komut başlığı
- Karta iletilecek mesaj

İkinci Adım:

Kart, gelen MAC_IN değerini doğrular. İşlem başarı ile sonlanmışsa kartta önceden belirlenmiş gizli anahtar (KEY_PROOF) kullanılarak işlem sinama değeri hesaplanır. Bu sinama değerinde aşağıdaki detaylar yer alır:

- Tag_Proof_Award
- İşlem Verisi: Sayaç veya kural tanımlayıcısı, terminal verisi, mevcut veri, işlem miktarı, kural yürütme durumu.
- Kart işlem sayacı (Komutun başında artırılır.)

Komuttan sonra, kartta ilgili komut için önceden tanımlanmış olan gizli anahtar kullanılarak MAC_OUT değeri hesaplanır. MAC_OUT'ta aşağıdaki alanlar bulunur:

- Tag_Mac_Award_Out
- CTC
- Komut Başlığı
- Terminale gönderilen mesaj

Üçüncü Adım:

Terminal, gelen MAC_OUT değerini doğrular. İşlem başarılı ise, işlem verisi ve sinama değeri operatöre iletilir.

Dördüncü Adım:

Operatör sinama değerini doğrular, başarılı ise işlem için onay verir.

İşlem Sinama Kuralları

İşlem sinama için aşağıdaki kurallara uyulması tavsiye edilmektedir:

- Terminal İşlem sinama değeri hesaplayamaz.
- İşlem sinama için kullanılacak gizli anahtarlar başka bir işte kullanılmamalıdır.
- Sayaç veya kural dosyasında belirtilen anahtar referansı 00h ise işlem sinama gerekli değildir. (Kart sıfır geri döndürür.)

Erişim Koşul Kontrolü

GemClub kartlarında, kullanıcı veya firmanın kart üzerinde belirli işlem yapıp yapamamasını belirlemek için erişim koşulları kullanılır. Bunlar sistemde dolaylı olarak tanımlanabilceği gibi (işletim sistemindeki bazı işlemlerde erişim koşulu kontrolü otomatik yapılır) ilgili bir dosya yaratılırken de verilebilir.

'ISO Kayıt Yönetim Komutları' için:

- İlgili dosya için PIN kodu ya da gizli kod içeren bir erişim koşulu tanımlanmış ise bu kodun 'ISO Kayıt' komutundan önce karta verilmesi gerekmektedir.
- Kod doğru olarak verilmişse '*Kod Doğrula*' komutu ile enel güvenlik durumu güncellenir.
- Erişim koşulu olarak güvenli mesajlaşma seçilmişse, Mesaj doğrulama kodu (MAC) da 'ISO Kayıt' komutu ile birlikte verilmelidir.

Diğer fonksiyonlar için:

- APDU komutunun yapısı dosyada belirtilen erişim koşuluna bağlı değildir. MAC ya da gizli kod belirtilmezse doğrulama verisi olarak sıfır gönderilir.
- Erişim koşulu olarak gizli kod gerekiyorsa, bu değer komutun veri alanında tutulabilir.
- Genel güvenlik durumu bu tip kod bildirimi için güncellenmez.

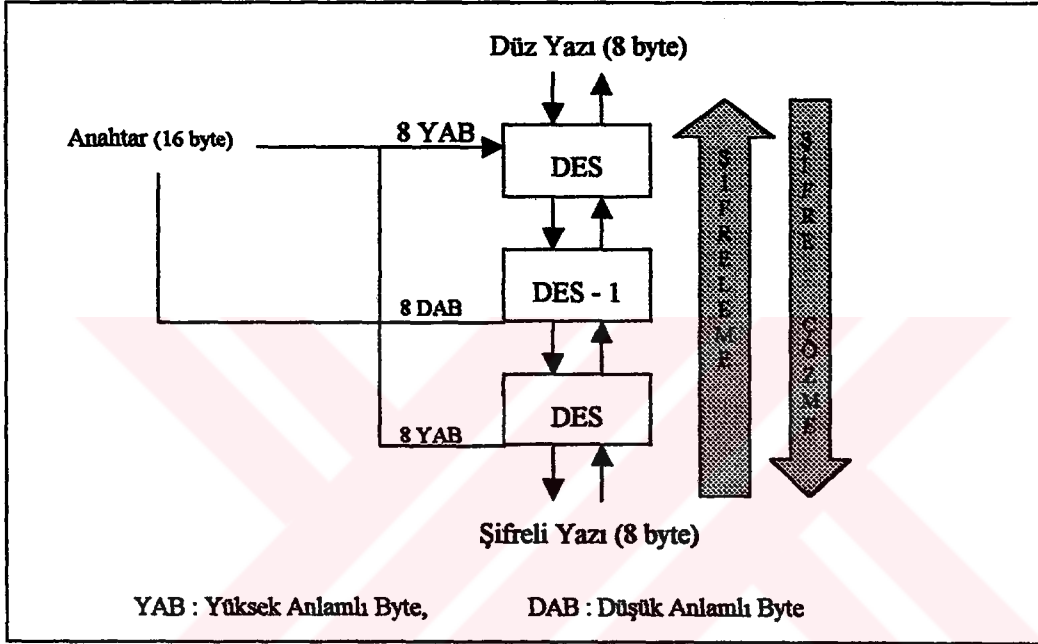
- Eriřim kořulunda PIN kod gerekli gsteriliyorsa komuttan nce ‘Kod Doęrula’ komutu yrtlecektir. Bu iřlem ile, PIN kodu iin kartın genel gvenlik durumu gncellenecektir.

4. GemClub řifreleme Yntemleri

Bu blmde, GemClub kartlarındaki gvenli mesajlařma ve kullanılan komutlara iliřkin mesaj doęrulama kodları ele alınacaktır.

GemClub’da Gvenli Mesajlařma – 3DES Algoritması

GemClub kartlarında gvenli mesajlařma ve řifreleme iin 3DES algoritması kullanılmaktadır. ift uzunluklu anahtarlar kullanılarak yrtlebilen 3DES algoritması ile řifreleme veya řifre zme řekil A.9’da verilmiřtir.



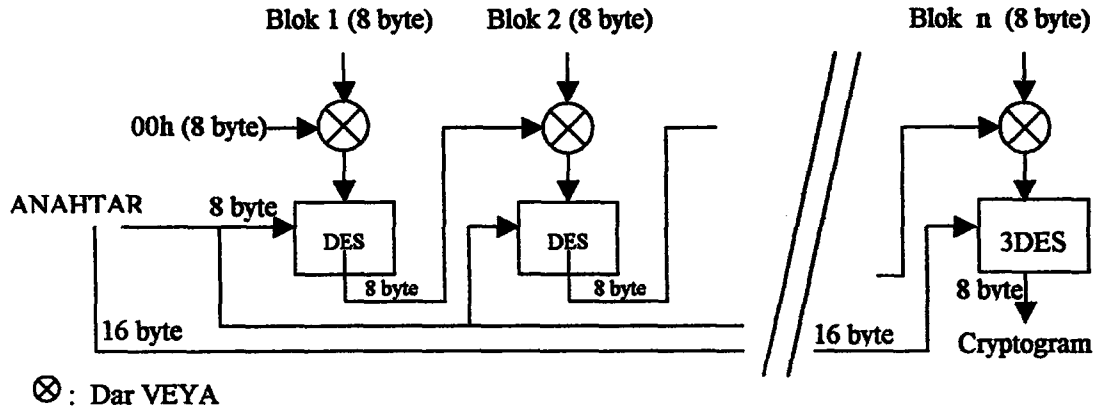
řekil A.9 3DES řifreleme / řifre zme Algoritması

MAC Hesaplaması

GemClub kartta, ařaęıdaki kořullara uyulması durumunda mesaj doęrulama kodu (MAC) CBC řifreleme algoritmasının son bloęunda hesaplanır.

- Ara bloklarzerinde yksek anlamlı 8 byte (pratikte, ilk yarım anahtar) kullanılarak DES uygulanır.
- Sonuncu blokta ise 16 byte’lık anahtar kullanılarak 3DES yrtlr.

MAC hesaplama algoritması řekil A.10’da gsterilmektedir.



Şekil A.10 MAC Hesabı

GemClub'da Şifreleme ve Komutlar

GemClub kartlarda işlemler için birkaç MAC değeri kullanılır. Bu MAC veri alanlarının içeriği Tablo A.21 – Tablo A.24'te verilmiştir.

MAC hesaplamasında kullanılan her bir bloğun başında yer alan bir byte uzunluğundaki tekil sayıya 'Etiket' adı verilir.

Ödül İşlemleri

Tablo A.21 Ödül işlemlerinde kullanılan MAC değerleri

MAC_IN

Ad	Kod
KEY (16 byte)	KEY_AWARD – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_IN_AWARD
MAC Verisi ve Doldurma	Blok 1 = (Etiket_{1 byte} CTC_{2 byte} Komut başlığı_{15 byte}) Blok 2 = (Etiket_{1 byte} Gelen Veri_{7 byte}) Blok 3 = (Etiket_{1 byte} Gelen Veri(devamı)_{7 byte}) Blok 4 = (Etiket_{1 byte} 80h 00 00 00 00 00 00)

İşlem Sınama

Ad	Kod
KEY (16 byte)	KEY_PROOF – Sınama hesabı için ilgili sayaçta adı geçen anahtar
Etiket (1 byte)	TAG_PROOF_AWARD
MAC Verisi ve Doldurma	Blok 1 = (Etiket_{1 byte} CTC_{2 byte} Sayaç Tanımlayıcısı_{1 byte} Terminal Verisi_{4 byte}) Blok 2 = (Etiket_{1 byte} Terminal Verisi (devamı)_{4 byte} Tarih₃)

	byte) Blok 3 = (Etiket ₁ byte İşlem Tutarı ₃ byte 80h 00 00 00)
--	--

MAC_OUT

Ad	Kod
KEY (16 byte)	KEY_AWARD – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_OUT_AWARD
MAC Verisi ve Doldurma	Blok 1 = (Etiket ₁ byte CTC ₂ byte Komut başlığı ₅ byte) Blok 2 = (Etiket ₁ byte Giden Veri ₇ byte) Blok 3 = (Etiket ₁ byte Giden Veri(devamı) ₆ byte 80h)

İndirim İşlemleri

Tablo A.22 İndirim işlemlerinde kullanılan MAC değerleri

MAC_IN

Ad	Kod
KEY (16 byte)	KEY_REDEEM – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_IN_REDEEM
MAC Verisi ve Doldurma	Blok 1 = (Etiket ₁ byte CTC ₂ byte Komut başlığı ₅ byte) Blok 2 = (Etiket ₁ byte Gelen Veri ₇ byte) Blok 3 = (Etiket ₁ byte Gelen Veri(devamı) ₇ byte) Blok 4 = (Etiket ₁ byte 80h 00 00 00 00 00 00)

İşlem Sınama

Ad	Kod
KEY (16 byte)	KEY_PROOF – Sınama hesabı için ilgili sayaçta adı geçen anahtar
Etiket (1 byte)	TAG_PROOF_REDEEM
MAC Verisi ve Doldurma	Blok 1 = (Etiket ₁ byte CTC ₂ byte Sayaç Tanımlayıcısı ₁ byte Terminal Verisi ₄ byte) Blok 2 = (Etiket ₁ byte Terminal Verisi (devamı) ₄ byte Tarih ₃ byte) Blok 3 = (Etiket ₁ byte İşlem Tutarı ₃ byte 80h 00 00 00)

MAC_OUT

Ad	Kod
KEY (16 byte)	KEY_REDEEM – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_OUT_REDEEM
MAC Verisi ve Doldurma	Blok 1 = (Etiket_{1 byte} CTC_{2 byte} Komut başlığı_{5 byte}) Blok 2 = (Etiket_{1 byte} Giden Veri_{7 byte}) Blok 3 = (Etiket_{1 byte} Giden Veri(devamı)_{6 byte} 80h)

Kural Kullanma İşlemleri

Tablo A.23 Kural Kullanma işlemlerinde kullanılan MAC değerleri

MAC_IN

Ad	Kod
KEY (16 byte)	KEY_USE_RULE – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_IN_USE_RULE
MAC Verisi ve Doldurma	Blok 1 = (Etiket_{1 byte} CTC_{2 byte} Komut başlığı_{5 byte}) Blok 2 = (Etiket_{1 byte} Gelen Veri_{7 byte}) Blok 3 = (Etiket_{1 byte} Gelen Veri(devamı)_{7 byte}) Blok 4 = (Etiket_{1 byte} 80h 00 00 00 00 00 00)

İşlem Sınama

Ad	Kod
KEY (16 byte)	KEY_PROOF – Sınama hesabı için ilgili sayaçta adı geçen anahtar
Etiket (1 byte)	TAG_PROOF_USE_RULE
MAC Verisi ve Doldurma	Blok 1 = (Etiket_{1 byte} CTC_{2 byte} Sayaç Tanımlayıcısı_{1 byte} Terminal Verisi_{4 byte}) Blok 2 = (Etiket_{1 byte} Terminal Verisi (devamı)_{4 byte} Tarih_{3 byte}) Blok 3 = (Etiket_{1 byte} İşlem Tutarı_{3 byte} Kural Yürütme Durumu_{1 byte} 80h 00 00)

MAC_OUT

Ad	Kod
----	-----

KEY (16 byte)	KEY_USE_RULE – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_OUT_USE_RULE
MAC Verisi ve Doldurma	Blok 1 = (Etiket _{1 byte} CTC _{2 byte} Komut başlığı _{5 byte}) Blok 2 = (Etiket _{1 byte} Giden Veri _{7 byte}) Blok 3 = (Etiket _{1 byte} Giden Veri(devamı) _{5 byte} 80h 00)

Diğer Güvenli Mesajlaşma İşlemleri

GemClub’da MAC kullanan diğer işlemler şunlardır:

- Parametre Güncelle
- Nesne Yarat
- Nesne Yoket
- Kayıt Güncelle
- Kayıt Ekle

Tablo A.24 Diğer işlemlerde kullanılan MAC değerleri

MAC_IN

Ad	Kod
KEY (16 byte)	KEY_UPDATE_PARAM – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_IN_UPDATE_PARAM
MAC Verisi ve Doldurma	Blok 1 = (Etiket _{1 byte} CTC _{2 byte} Komut başlığı _{5 byte}) Blok 2 = (Etiket _{1 byte} Gelen Veri _{değişken}) ... Blok n = (Etiket _{1 byte} Gelen Veri(devamı) _{değişken} Doldurma)

MAC_OUT

Ad	Kod
KEY (16 byte)	KEY_UPDATE_PARAM – İlgili erişim koşulunda adı geçen anahtar
Etiket (1 byte)	TAG_MAC_OUT_UPDATE_PARAM
MAC Verisi ve Doldurma	Blok 1 = (Etiket _{1 byte} CTC _{2 byte} Komut başlığı _{5 byte}) Blok 2 = (Etiket _{1 byte} Giden Veri _{2 byte} 80h 00 00 00 00)

MAC Etiket Yönetim Kuralları

GemClub kartlarında Tablo A.25’de verilen önceden tanımlı etiketler kullanılmaktadır.

Tablo A.25 Önceden Tanımlanmış Etiket Değerleri

Komut	Etiket
TAG_MAC_IN_AWARD	11h
TAG_MAC_OUT_AWARD	12h
TAG_PROOF_AWARD	13h
TAG_MAC_IN_REDEEM	21h
TAG_MAC_OUT_REDEEM	22h
TAG_PROOF_REDEEM	23h
TAG_MAC_IN_USE_RULE	31h
TAG_MAC_OUT_USE_RULE	32h
TAG_PROOF_USE_RULE	33h
TAG_MAC_IN_UPDATE_PARAM	51h
TAG_MAC_OUT_UPDATE_PARAM	52h
TAG_MAC_IN_CREATE	61h
TAG_MAC_OUT_CREATE	62h
TAG_MAC_IN_DELETE	71h
TAG_MAC_OUT_DELETE	72h
TAG_MAC_IN_UPDATE_REC	81h
TAG_MAC_OUT_UPDATE_REC	82h
TAG_MAC_IN_APPEND_REC	91h
TAG_MAC_OUT_APPEND_REC	92h

Doldurma Kuralları

GemClub’da şifreleme yöntemi olarak 3DES algoritması kullanıldığından şifrelenecek veya şifresi çözülecek olan verilerin uzunluğu 8 byte’ın katı olmalıdır. Eğer gönderilen veya alınan verinin boyu bu şartı sağlamıyorsa kalan boşluklar ‘doldurulur’. Bu işlem aşağıda belirtilen kurallara göre gerçekleştirilir.

- Şifrelenecek veya şifresi çözülecek olan veri, 3DES algoritmasında belirtilen kurallara uymalıdır. Boyu 8 byte'ın katı olan bloklar halinde iletim yapılmalıdır.
- MAC hesaplaması sırasında, değeri 80h olan en az bir byte verinin sonuna eklenmelidir. Oluşan blok 8 byte'ın katı değilse sağdan sıfırlar ile doldurulmalıdır.

Örnekler:

- MAC uzunluğu 13 byte ise, sağdan 80 00 00h eklenerek toplam 16 byte'lık uzunluğa ulaşılır.
- MAC uzunluğu 15 byte ise, sağdan 80h eklenerek toplam 16 byte'lık uzunluğa ulaşılır.
- MAC uzunluğu 16 byte ise, sağdan 80 00 00 00 00 00 00h eklenerek toplam 24 byte'lık uzunluğa ulaşılır.

5. Diğer GemClub Özellikleri

Bu bölümde, GemClub kartlarındaki yedekleme yordamı ve nokta çevrim özellikleri ele alınacaktır.

GemClub Yedekleme Yordamı

Önemli verilerin iletim sırasında veya güç kesintisi sonrası değişmesi veya bozulmasını önlemek için 'Yedekleme Yordamı' özelliği kullanılır. Bu özellik sayesinde, hataya düşmüş bir işlemten sonra, işlem öncesi yedeklenen bilgilere geri dönülebilir. Bir komut bloke olduğunda ya da bir dosya yanlışlıkla bozulduğunda etkin hale geçen bu özellik, komut öncesi saklanan verilere geri dönülür.

Aşağıdaki komutların kullanımlarında yedekleme özelliği otomatik olarak etkinleşir:

- Nesne Yarat/Yoket
- Parametre Güncelle
- Ödül/İndirim
- Kural Kullan
- Sistem verilerinin güncellendiği tüm işlemler. (sınama değeri, dosya tanımlayıcısı, vb.)

Dosya güncellemelerinde, 'Yanıtı Al' komutu başarı ile yürütüldükten sonra dosya yedeklenmesi işlemi tamamlanmış olur.

Kural Mekanizması – Puan Dönüşümleri

GemClub kartlarındaki Kural Yordamı kullanılarak sayaçlara puan kazandırma veya sayaçlardan puan düşme işlemleri gerçekleştirilebilir. Belirli bir kuralın işlenmesi sırasında yürütülmek üzere en fazla dört makrokomut tanımlanabilir. Her bir kural farklı dört sayaç üzerinde tanımlanabilir.

Makrokomutlarda aşağıdaki bilgiler mutlaka bulunmalıdır:

- Kullanılacak sayacın tanımlayıcısı
- Yürütülecek komut tipi (Ödül veya İndirim)
- Parametre A
- Parametre B

'Puan Dönüşümlü Kural Kullan' komutu tanımlandığında, bu komutla ilgili bir 'Kural Kullan' komutu geldiğinde GemClub otomatik olarak, gelen verileri verilecek veya geri alınacak puanlara dönüştürür.

A ve B parametreleri, sayaç üzerinde tanımlanacak puan dönüşümünün belirtilmesinde kullanılırlar.

Parametre A : Sayaç değeri Parametre B'ye ulaştığında sayaçtan çıkarılacak veya sayaca eklenecek puan sayısını verir.

Parametre B : Sayaç değerinde değişiklik yapılması için ulaşılması gereken değeri belirtir. 'Kural Kullan' komutunda belirtilen işlem miktarı ile aynı birimden olmalıdır.

GemClub'da kullanılan puan dönüşüm formülü şöyledir:

$$Puan = Parametre_A * \left\| \frac{Miktar}{Parametre_B} \right\| \quad (A.1)$$

Örneğin, 0Dh sayacına uçulan her kilometre için 50 puan verilmesi durumunda, A = 50 ve B = 1000'dir. Kartsahibi 4990 km uçmuş ise,

$$Puan = 50 * [4990 / 1000] = 200$$

puan kazanacaktır.

B parametresi sıfır olarak tanımlanırsa, miktar ne olursa olsun her işlemde A parametresi kadar puan verilir.

6. GemCLub Komut Formatı

Bu bölümde, GemClub kartlarında kullanılan Uygulama Protokol Veri Birimi (APDU - Application Protocol Data Unit) ele alınacaktır.

GemCLub'da, ISO 7816-4 standardında tanımlanan APDU formatında komut ve yanıtlar geçerlidir. Böylece, GemCLub komutlarının GemPlus GCR arayüz sürücü kütüphanesi ile uyumu sağlanmış olur. GemClub iletim (transport) katmanı, ISO 7816-3 standardında belirtilen T = 0 protokolü ile uyumludur. Burada APDU, iletim katmanı veri birimi 'ne (TPDU - Transport Protocol Data Unit) çevrilir. Okuyucu ile kart arasında TPDU bazında haberleşme yapılır.

GemClub kartlarda aşağıda belirtilen komutlar kabul edilir:

- Komut ya da yanıt yok: Sıfır uzunluklu bir T=0 ISO IN TPDU olarak gönderilir.
- Kısa Format: Komut yok, ama 1 ila 256 byte arası bir yanıt mevcut. Bu, T=0 ISO OUT TPDU olarak gönderilir.
- Kısa Format: 1 ila 255 byte arası uzunlukta komut verisi; yanıt yok. T=0 ISO OUT TPDU olarak gönderilir.
- Kısa Format: 1 ila 255 byte arası uzunlukta komut verisi; 1 ila 256 byte arası bir yanıt mevcut. Komut, T=0 ISO IN TPDU olarak gönderilir. Ardından T=0 ISO OUT TPDU olarak gönderilen bir 'Yanıtı Al' komutu yollanmalıdır. Yanıt alma mekanizması ISO 7816-4 standardı ile uyumludur.

Ek. B Veritabanı Yapısı**Tablo Adı : TBL_BINA**

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
BinaNo	int	4	HAYIR
Ad	char	50	EVET
KampusNo	int	4	EVET

Tablo Adı : TBL_BOLUM

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
BolumNo	int	4	HAYIR
Ad	char	50	EVET
FakulteNo	int	4	EVET
BaskanSicilNo	numeric	18	EVET
Bolum	bit	1	HAYIR

Tablo Adı : TBL_DERS

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
DersNo	int	4	HAYIR
Ad	char	100	EVET
BolumNo	int	4	EVET
DersTipNo	tinyint	1	EVET
OgrGrvSicilNo	numeric	18	EVET
Donem	tinyint	4	HAYIR
Saat	tinyint	1	EVET
Kredi	tinyint	1	EVET
Aciklama	varchar	500	EVET
ArrSicilNo	numeric	18	EVET

Tablo Adı : TBL_DERSPRG

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
KayitNo	numeric	18	HAYIR
Yil	char	4	EVET
YariYil	char	1	EVET
DersNo	int	4	HAYIR
Gun	tinyint	1	EVET
Seans	tinyint	1	EVET
BinaNo	int	4	EVET
SinifNo	int	4	EVET

Tablo Adı : TBL_FAKULTE

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
FakulteNo	int	4	HAYIR
Ad	char	50	EVET
BinaNo	int	4	EVET
DekanSicilNo	numeric	18	EVET
Fakulte	bit	1	HAYIR

Tablo Adı : TBL_HATALAR

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
HataNo	int	44	HAYIR
Aciklama	varchar	100	HAYIR

Tablo Adı : TBL_KAMPUS

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
KampusNo	int	4	HAYIR
Ad	char	30	EVET
Semt	char	20	EVET
il	char	20	EVET

Tablo Adı : TBL_KULLANICI

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
KullaniciNo	numeric	18	HAYIR
UnvanNo	char	10	EVET
Ad	char	30	EVET
Soyad	char	30	EVET
GirisYili	char	4	EVET
ABDaliNo	int	4	EVET
BolumNo	int	4	EVET
Adres	varchar	300	EVET
TelefonNo	char	10	EVET
EMail	char	50	EVET
Resim	char	20	EVET
imza	char	20	EVET
Cinsiyet	char	1	EVET
Uyruk	char	3	EVET
NufusSeriNo	char	15	EVET
BabaAdi	char	30	EVET
AnneAdi	char	30	EVET
DogumTarihi	char	8	EVET
DogumYeri	char	20	EVET
MedeniHali	char	1	EVET
Dini	char	10	EVET
il	char	20	EVET
ilce	char	20	EVET
Mahalle	char	20	EVET
CiltNo	char	10	EVET
AileSiraNo	tinyint	1	EVET
SiraNo	tinyint	1	EVET
DahiliNo	char	4	EVET

OdaNo	int	4	EVET
Sifre	char	8	EVET
KurumNo	it	4	EVET

Tablo Adı : TBL_OGRDERS

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
KayitNo	numeric	18	HAYIR
OgrenciNo	numeric	18	EVET
Yil	char	4	EVET
YariYil	char	1	EVET
DersNo	int	4	EVET
Vize1	tinyint	1	EVET
Vize2	tinyint	1	EVET
Final	tinyint	1	EVET
Odev1	tinyint	1	EVET
Odev2	tinyint	1	EVET
Odev3	tinyint	1	EVET
Odev4	tinyint	1	EVET
Quiz1	tinyint	1	EVET
Quiz2	tinyint	1	EVET
SOrtalama	tinyint	1	EVET
KOrtalama	char	2	EVET
KatilimSaat	tinyint	1	EVET
Hak	tinyint	1	EVET
Durum	tinyint	1	EVET

Tablo Adı : TBL_ONKOSUL

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
DersNo	int	4	HAYIR
KosulDersNo	int	4	HAYIR

Tablo Adı : TBL_PARAM

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
DersKayitBasTar	char	8	EVET
DersKayitBitTar	char	8	EVET
BaslangicTarihi	char	8	EVET
BitisTarihi	char	8	EVET
FinalBasTar	char	8	EVET
FinalBitTar	char	8	EVET
YariYil	char	1	EVET
Yil	char	4	EVET

Tablo Adı : TBL_ROL

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
ROL	varchar	20	HAYIR
KOD	varchar	10	HAYIR

Tablo Adı : TBL_ROLEKRAN

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
ROL	varchar	20	HAYIR
EKRAN	varchar	20	HAYIR

Tablo Adı : TBL_SABIT

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
TanimNo	int	4	HAYIR
Alan	varchar	50	HAYIR
Deger	tinyint	1	HAYIR
Aciklama	varchar	50	HAYIR

Tablo Adı : TBL_SINAVPRG

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
KayitNo	numeric	18	HAYIR
Yil	char	4	EVET
YariYil	char	1	EVET
DersNo	int	4	EVET
Tarih	char	8	EVET
Seans	tinyint	1	EVET
SinavTipNo	tinyint	1	EVET
BinaNo	int	4	EVET
SinifNo	int	4	EVET

Tablo Adı : TBL_SINIF

Alan Adı	Veri Tipi	Uzunluk	NULL Olabilir mi?
SinifNo	int	4	HAYIR
Ad	char	50	EVET
BinaNo	int	4	EVET
Kapasite	tinyint	1	EVET
SinifTipNo	tinyint	1	EVET

Sabitler

Alan	Değer	Açıklama
DERSTIPNO	10	LİSANS -SC
DERSTIPNO	11	LİSANS -ZR
DERSTIPNO	20	Y.LİSANS-SC
DERSTIPNO	21	Y.LİSANS-ZR
DERSTIPNO	30	DOKTORA - SC
DERSTIPNO	31	DOKTORA -ZR
DONEM	1	1
DONEM	2	2
DONEM	3	3
DONEM	4	4
DONEM	5	5
DONEM	6	6
DONEM	7	7
DONEM	8	8
GUN	1	PAZARTESİ

GUN	2	SALI
GUN	3	ÇARŞAMBA
GUN	4	PERŞEMBE
GUN	5	CUMA
SEANS	1	09:00-09:50
SEANS	2	10:00-10:50
SEANS	3	11:00-11:50
SEANS	4	12:00-12:50
SEANS	5	13:00-13:50
SEANS	6	14:00-14:50
SEANS	7	15:00-15:50
SEANS	8	16:00-16:50
SEANS	9	17:00-17:50
SINAVTIPNO	0	VİZE
SINAVTIPNO	1	FINAL
SINAVTIPNO	2	QUIZ
SINIFTIPNO	0	ODA
SINIFTIPNO	1	SINIF
SINIFTIPNO	2	LAB.
UNVANNO	0	DİĞER
UNVANNO	1	KURUM
UNVANNO	2	LİSANS
UNVANNO	3	Y.LİSANS
UNVANNO	4	DOKTORA
UNVANNO	5	AR.GÖR.
UNVANNO	6	DR.
UNVANNO	7	YRD.DOÇ.DR.
UNVANNO	8	DOÇ.DR.
UNVANNO	9	PROF.

Ek C. Kontaklı Kartlarda Standart İletişim Protokolleri

ISO standardı, Tümlleşik Devreli Kart ile arabirim cihazı arasında veri alışverişi için tek bir hat tanımlamaktadır. Dolayısıyla, kartın veri gönderiyor veya alıyor olması durumlarına göre hattın iletim yönünün değişmesi gerekmektedir. Pratikte ikisinin gerçek anlamda eş zamanlı meydana gelmesi mümkün değildir; bu nedenle de 'Hat Geri Dönüş Zamanı' (line turnaround time) modem dünyasında sıkça rastlanan bir terimdir. İletim protokolu hattın çift yönlü iletim yapabilmesine duyulan ihtiyacı gözönünde bulundurmalıdır [9].

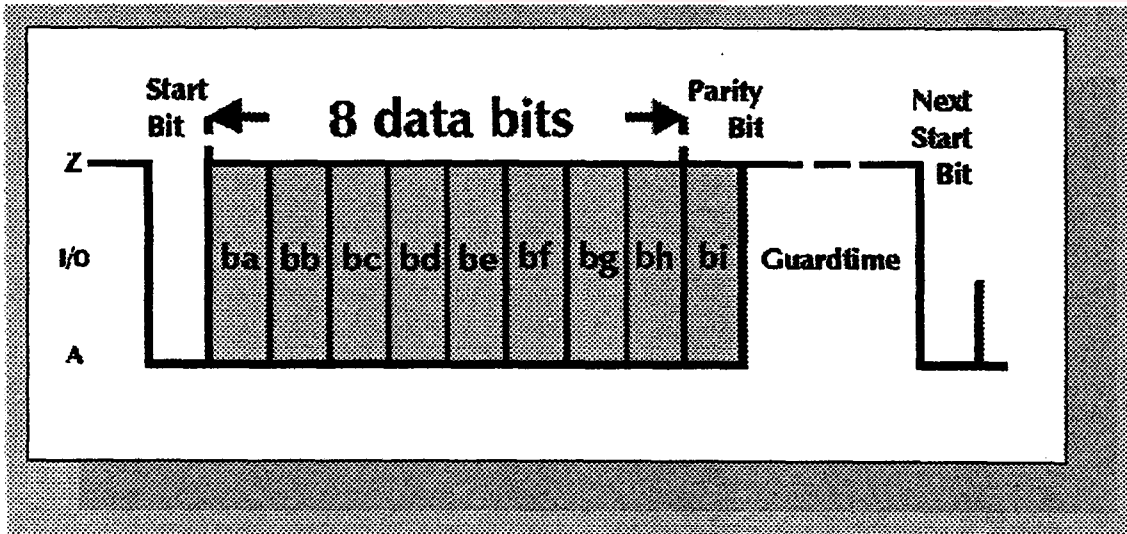
Karakter İletimi

Mikroişlemcili bir çok tümlleşik devre kartının sunduğu iletim karakteristikleri genellikle asenkron yarı çift yönlü çalışma şekillerine dayanmaktadır. T=1 protokolü blok şeklinde işlemi tanımlarken, T=0 iletişim protokolünde bu, sekizlilerin iletimini de kapsamaktadır. Seri iletişim, gönderme işlemini kartın veya arabirim cihazının yapıp yapmadığına bağlı olarak veri iletim yönünün değiştiği tek bir kırkuk konnektörü üzerinden kotalırlmaktadır. Bu yarı çift yönlü iletişim olarak adlandırılmaktadır. Tam çift yönlü iletişimde ise her iki yönde eşzamanlı gerçekleşen iletim için iki giriş/çıkış konnektörü kullanılmaktadır [9].

Asenkron iletişim tipi kişisel bilgisayarlarda karşılaşılan RS232C konnektörleri tarafından kullanıllana benzerdir. Kişisel bilgisayarlarda tam çift yönlü iletişim olmasına rağmen bir byte'lık tek bir karakterin iletimi için aşağıdaki gibi birkaç bitin fazladan eklenmesine ihtiyaç vardır:

- Başlangıç biti: Karakter çerçevesi senkronizasyonu için kullanılır.
- Parite biti: Hata yakalamak için kullanılır.
- Koruma Zamanı (Guardtime) : İki karakteri birbirinden ayırmak için kullanılır.

Bir karakter çerçevesinin formatı Şekil C.1 'de verilmiştir.



Şekil C.1. Karakter Çerçevesi

Alıcı, giriş/çıkış arabirimini sürekli kontrol ederek işaretten boşluğa veya yüksek gerilimden düşük gerilime geçişleri yakalamaya çalışır. Hattın örneklenme periyodu

öyle bir değer almalıdır ki, alıcının örnek alma anı her bit dizisinin en fazla %20'lik bir hata payıyla ortasına denk gelecek şekilde olmalıdır. Parite bitinin, çift parite yöntemine uygun olarak kullanılması gerektiği tanımlanmıştır. Bu, sekiz bitlik veri ile parite bitinin içindeki 1'lerin sayısının çift olması anlamına gelmektedir.

Koruma zamanı (blok modunda bir bitlik periyot uzunluğunda olacak şekilde değiştirilebilmesine rağmen) iki bitlik periyot uzunluğunda olacak şekilde tanımlanmıştır. Bu, bilgisayarlarda kullanıldığı gibi UART'da (Universal Asynchronous Receiver Transmitter) iki durma bitine sahip olmaya benzer.

Albaştan sürecindeki asenkron seri iletimin daha genel tanımı; 9600 bit/saniye, 8 veri biti, çift parite ve yarı çift yönlü çalışmada iki durma biti olarak verilebilir. "Yarı çift yönlü iletişim" terimi sadece, bir bilgisayarda da UART ile mükemmel olarak gerçekleştirilebildiği gibi, bir anda yalnızca bir tek yöne doğru veri yollama şekli için kullanılmıştır. Bununla birlikte, RS232C arabirimi veri yollama ve alma için iki farklı bağlantı kullanır. Dolayısıyla, tek bağlantı içeren tümleşik devre kartı ile doğrudan haberleşebilmesi için donanımının değiştirilmesi gerekir.

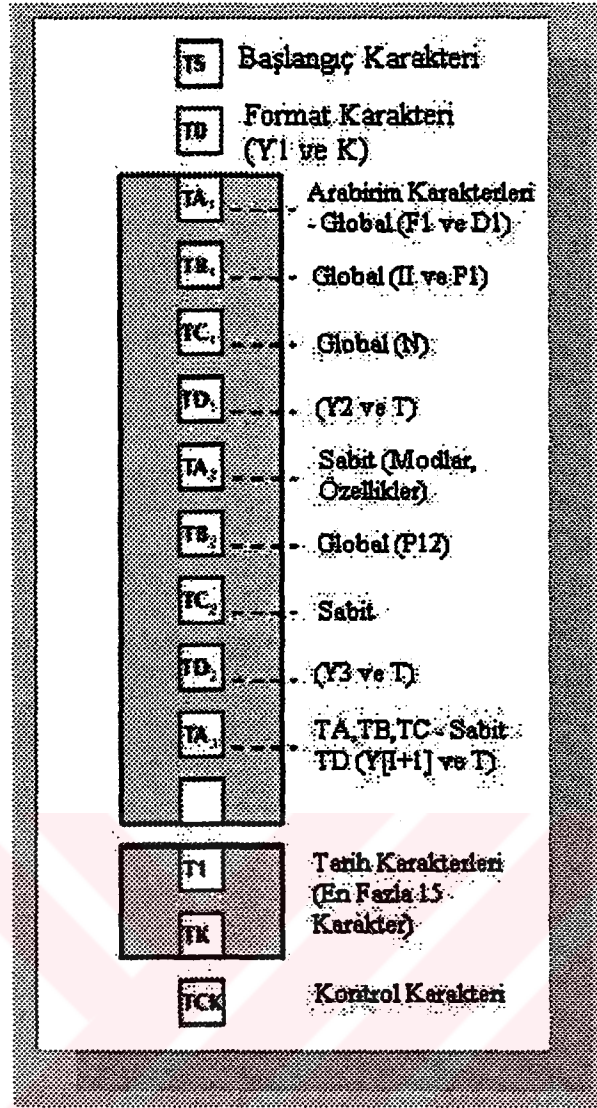
Asenkron karakter yollamanın, bilgisayarı bir arabirim cihazı gibi davranmaya zorlamak gibi başka problemleri de vardır. 7816-3 standardı, normal bir bilgisayar UART'ı ile gerçekleştirilemeyecek ama $T=0$ 'da zorunlu olan bir hata yakalama ve düzeltme mekanizması tanımlar. Alıcı alış sırasında herhangi bir parite hatasına rastladığında, ilk dur bitinin tam ortasında, giriş/çıkış hattını yüksek empedans durumuna ya da düşük seviyeye getirir. Verici ise, ikinci dur biti periyodunun başında giriş/çıkış hattını örneklemekle yükümlüdür. Bir hata durumu algılandığında verici hatalı alınan karakteri tekrar yollamak zorundadır. Tabi ki verici dur bitlerini çıkış olarak veremez ama koruma zamanı boyunca hattın durumunu algılayabilmek için hattın yüksek seviyeye çıkmasına izin vermelidir. Tümleşik devre kartı ve arabirim cihazı için verilen bu sıkı ilişkinin, bu seviyedeki hata kontrolünün dezavantajlarının yanında daha fazla yarar sağlayıp sağlamayacağı sorgulanmalıdır. OSI modelindeki daha yüksek seviyeli hata kontrolü bu durum için daha çok tercih edilmektedir ve bunu uygulama seviyesinde gerçekleştirmesine rağmen, $T=1$ haberleşme protokolü çerçeve seviyesinde hata kontrolü uygular.

Albaştan Yanıtı

Albaştan sinyali arabirim cihazı tarafından uygulandıktan sonra, tümleşik devre kartı albaştan yanıtıyla, karşılık verir. Aktif ve düşük albaştan modu için, tümleşik devre albaştan sinyalinin artan kısmı için 400 ile 40000 saat devri arasında karşılık vermelidir. Albaştan yanıtı, ilk karakteri de içine alarak maksimum 33 karakterdir ve beş alandan oluşur.

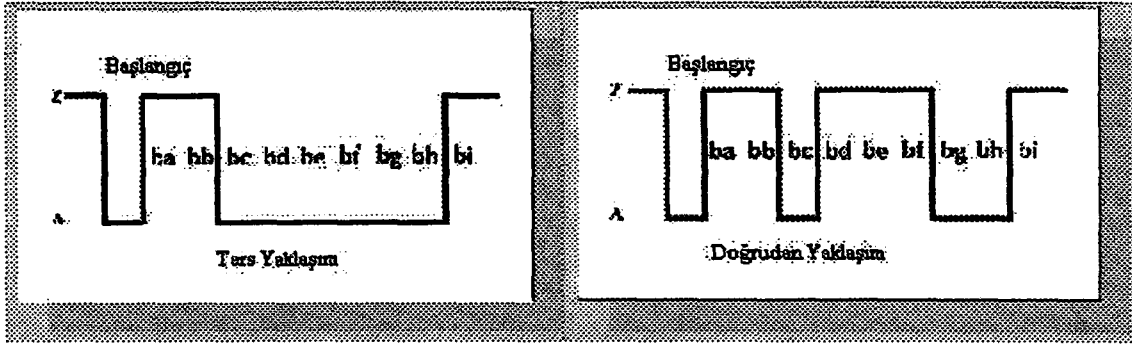
- Başlangıç karakteri (TS)
- Format karakteri (TO)
- Arabirim karakterleri (TA_i, TB_i, TC_i, TD_i)
- Tarih karakterleri ($T_1, T_2, \dots, TK$)
- Kontrol karakteri (TCK)

Bu alanlardan herbiri Şekil C.2' de görüldüğü sıra ile gönderilir.



Şekil C.2. Albastan Yanıtı

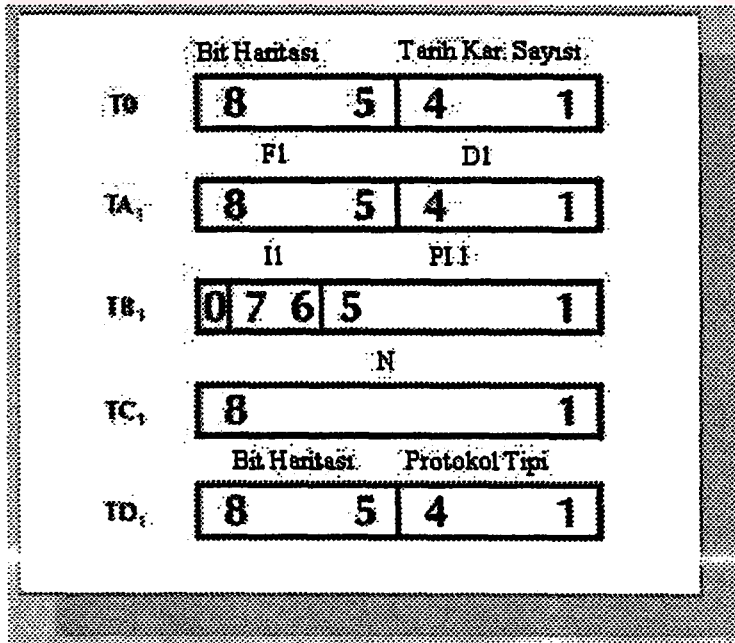
Başlangıç karakteri TS, gerçekte veri iletim hızına (otomatik baud hızı algılama) ve ayrıca devrenin duyarlılığına karar vermek amacıyla senkronizasyon için kullanılır. TS karakterinin formatı Şekil C.3'de gösterilmiştir. Şekil iki olasılığı da, doğrudan ve ters yaklaşım, göstermektedir. Lojik seviyesinin 1 olduğu ters yaklaşımda, boşluk ya da en anlamlı bit ilk olarak gönderilir. Lojik seviyesinin 1 olarak işaretlendiği ya da üst seviyede olduğu doğrudan yaklaşım ile en küçük anlamlı basamak ilk olarak gönderilir. Uygun lojik algılama, başlangıç karakterinin, ters yaklaşım için 3F, doğrudan yaklaşım için ise onaltılık düzende 3B olarak değerlendirilmesiyle sonuçlanır.



Şekil C.3. Başlangıç Karakteri

Format karakteri TO, albaştan karakterlerine kalan yanıtı çözmesi için gerekli bilgiyi sağlar. En yüksek anlamlı 4 bit TA1, TB1, TC1 ve TD1 in var olup olmadığını bildirmek için bir bit haritası kullanılır. Örneğin, en yüksek anlamlı bit (b8) yerleştirilirse, TD1 arabirim karakter alanında yer alır. Benzer olarak TC1'in bulunması b7'nin varlığıyla bildirilir. TO format karakterinin en düşük anlamlı 4 bit tarih alanındaki bitlerin sayısını ikili düzende verir. 4 bitin kullanımı tarih karakterleri alanının maksimum büyüklüğünü 15 bit olarak sınırlar. Arabirim karakterleri (TAi, TBi, TCi, TDi) albaştan yanıtının karmaşık kısmıdır. Programlama gerilimi ve EPROM'un halihazırdaki parametreleri ile birlikte, varolan iletişim protokollerine ilişkin bilgi taşırlar. Mevcut belirsizlikleri gidermek, protokol tipi ve protokol parametrelerini değiştiren operasyonun etkili bir yöntemle yapıldığından emin olmak için ISO 7816-3 de değişiklik önerisinde bulunulmuştur. Karmaşıklığın büyük kısmı, T=0 iletişim protokolünün ticari uygulamalarla geriye dönük uyumunu başarmayı arzulamaktan gelir. Şu anda, çoklu protokol işlemleri az olduğundan ya T=0 ya da T=1 iletişim protokolünü kullanan ticari uygulamalar vardır.

Bulunmaları zorunlu olmayan arabirim byte'ları Şekil C.4'te tanımlanmıştır. T0 ve TDi karakterleri TAi, TBi, TCi, and TDi byte'larının var olup olmadığını belirten bit haritalarına sahiptirler. TA1, TB1, TC1 ve TB2 karakterleri global arabirim byte'ları olarak tanımlanırlar ve kartın çalışması için temel teşkil ederler.



Şekil C.4. Zorunlu Olmayan Arabirim Byte'ları

TA1 seri iletimin temel karakterlerini tanımlar, FI saat frekansı tanımlama faktörüdür ve DI bit frekansı ayarlama faktörüdür. İkili düzendeki alanlar aşağıdaki gibi tanımlanan F ve D'nin gerçek değerlerine ulaşmak için standart tarafından verilen tablolarla karşılaştırılır.

İş (etu – elementary time unit) = $(1 / D) \times (F / f) \text{ sn}$
(2.1)

Temel zaman birimi (etu) karakter çerçevesinde kullanılan nominal bit süresidir. Böylece, daha önce tanımlandığı gibi, bir karakter çerçevesi 12 etu'ya eşittir. (1 başlangıç etu, 8 veri etu, 1 parity etu, 2 koruma zamanı etu).

F1 ve D1'in varsayılan değerleri 1 olmakla beraber, tablolarda F için 372, D için ise 1 değeri tanımlanmıştır. Sonuç olarak iş ve başlangıç etu değeri aynı şeylerdir. Bu varsayılan değerler için saat frekansı 1 MHz-5MHz arasında olmalıdır. TB1, EPROM programlama gerilimi ve akımını tanımlamak için kullanılır. II ve PI1 değerleri, I mA ve P volt değerlerini elde etmek için kullanılır. TB2'nin programlama gerilimini yüksek taneciklilikle (5 yerine 8 bit) kullanıldığına dikkat edilmelidir. TC1, ardışık karakterler arasındaki fazladan koruma zamanını tanımlamak için kullanılan N değerini sağlar. N, 0-254 etu aralığında olabilir. N'nin, 255'e eşit olması, minimum koruma zamanının kullanılması gerektiğini işaret eder (T=0 için 2 etu ve T=1 için 1 etu). Daha önce belirtildiği gibi T=0 iletişim protokolü, parite hatasının bulunması ve sinyallemenin uygulanması için fazladan koruma zamanı gerektirir.

TD1, TDI protokol tipinin 0 ve 15 arasında olduğunu gösterir,

- T = 0 Asenkron yarı çift yönlü byte iletimi
- T = 1 Asenkron yarı çift yönlü blok iletimi
- T = 2/3 Tam çift yönlü iletim için ayrılmış
- T = 4 Geliştirilmiş yarı çift yönlü bit iletimi için ayrılmış
- T = 5..13 Daha ileriki kullanım için ayrılmış (RFU)
- T = 14 ISO olmayan protokoller
- T = 15 İlerideki eklemeler için ayrılmış

Japonya'nın, ulusal blok asenkron protokolü için T=14'u kullandığı bilinmelidir.

TD1 byte'ı ayrıca, TA2, TB2, TC2 ve TD2'nin var olup varolmadığını bildiren bir bit haritası ihtiva eder.

Önerilen değişiklik, iletişim protokollerinin ve parametrelerinin seçiminde önemli bir rolü olan TA2 arabirim byte'ı için yeni bir kullanımı tanımlamaktadır.

Tarih Karakterleri

Tarih karakterleri, kartın kullanım ömrü ile ilgili bilgiyi taşımak için kullanılabilir. Aslında diğer bazı olasılıklar da bulunmakla beraber, bu karakterlerin kullanımı halen anlaşmaya tabidir. Bu konu ISO 7816'nin dördüncü kısmının bir parçası olarak daha değerlendirilecektir.

Kontrol Karakteri

Kontrol karakteri sadece albaştana yanıt olarak T=0 protokolü verildiğinde gönderilmelidir. Diğer bütün durumlarda kontrol karakteri ATR'nin son karakteri olarak yollanır. Kontrol karakteri, T0'dan sıfır olan kontrol karakterine kadar olan bütün bitlerin DARVEYA'sı alınarak hesaplanır.

Haberleşme Protokolleri

Halen genel kullanımda olan iki haberleşme protokolu bulunmaktadır,

- T=0 asenkron yarı çift yönlü karakter iletimi ve
- T=1 asenkron yarı çift yönlü blok iletimi

T=0 protokolü Fransa'da yaygın olarak kullanılan bir protokoldür ve aynı zamanda ISO 7816-3'de tanımlanmış olan tek protokoldür. 1992'de ISO, T=1 protokolünü "ISO 7816-3 'de Değişiklik 1" olarak standardize etmiştir. Tümleşik devre kartının ve arabirim cihazının ortak bir protokolle çalışmak durumunda olduğu açıktır. Son bir kaç yılda yapılan tartışmaların çoğunun amacı, en iyi konfigürasyona sahip ortak bir yöntemle ulaşmak olmuştur. Bu ilke protokol tipinin seçilmesi suretiyle belirlenmeye çalışılmıştır. Aslında bu, arabirim cihazının albaştana cevap verdikten sonra tümleşik devre kartına yolladığı özel bir komuttur. Sadece T=0 protokolü ile haberleşebilen mevcut ticari sistemlerle geriye uyumluluğu sağlayabilmek için orjinal ISO 7816-3 standardında bazı değişiklikler yapmak gereklidir.

İki çalışma modu ilkesini tanımlayan yeni bir kavram teklif edilmiştir:

- Uzlaşma Modu
- Sabit Mod

Uzlaşma modunda çalışan bir tümleşik devre kartının protokolü bir protokol tipi seçme komutu ile değiştirilebilir. Sabit modda çalışan bir tümleşik devre kartı protokol seçme komutunu kabul edemez ama bir albaştan komutu ile uzlaşma moduna sokulabilir. Tümleşik devre kartı, arabirim cihazına kendisinin uzlaşma moduna geçebileceğini gösterse bile piyasada bulunan cihazlar bu değişikliğin yapılabileceğinden habersiz olabilirler ve bu sebeple albaştan yapmaya hazırlanmayabilirler.

b8	b7	b6	b5	b4	b1
b8 : Mod Değiştirme Yeteneği 0 : Değiştirebilir, 1 : Değiştiremez b7-6 : (İlerki Kullanımlar İçin Saklanmış) b5 : Doğrudan/Dolaylı Parametreler 0 : Doğrudan, 1 : Dolaylı b4-1 : Protokol					

Şekil C.5. TA2 Arabirim Byte'ı

Çok protokollü bir kartın uzlaşma modunun, önceliği T=0 haberleşme protokolüne vermesine dikkat edilmelidir. Bir başka deyişle eğer T=0 protokolü varsa, albaştana yanıt bu varsayılan protokolün gerektirdiği yanıt olmalıdır. Albaştan yanıtının bir parçası olan TA2 arabirim byte'ı uygun protokol seçimine müsaade ederek gerekli bilgiyi verir. Bu bitin varlığı durumunda kodlanması Şekil C.5'de gösterilmiştir. Bu bitin olup olmaması aşağıda belirtilen çalışma modunun belirlenmesi için kullanılır.

- ATR içinde TA2 varsa – Sabit mod
- ATR içinde TA2 yoksa – Uzlaşma mod

Görülebileceği üzere TA2 byte'ı içindeki 8. bit arabirim cihazına kartın uzlaşma moduna girip giremeyeceğini söyler.

Protokol Tipi Seçimi (PTS)

Protokol tipi seçimi, arabirim tarafından iletişim protokolünü veya FI ve DI için ilk değerleri değiştirmek için kullanılır. PTS komutu, albaştan cevabının hemen ardından çalıştırılmalıdır ve sadece tümleşik devre kartı uzlaşma modunda iken uygun olur.

Arabirimi iki şekilde çalıştırmak mümkündür: Albaştana gelen cevap sonrası ilk başlatılan protokolü kullanarak ya da F ve D için varsayılan değerleri kullanarak. Bu, protokol ve iletişim parametrelerinin doğrudan seçimini sağlar. Arabirim bu durumda herhangi bir değişim yapmak istediğinde PTS komutunu çağırmalıdır.

PTR isteği aşağıdaki parçalardan oluşur:

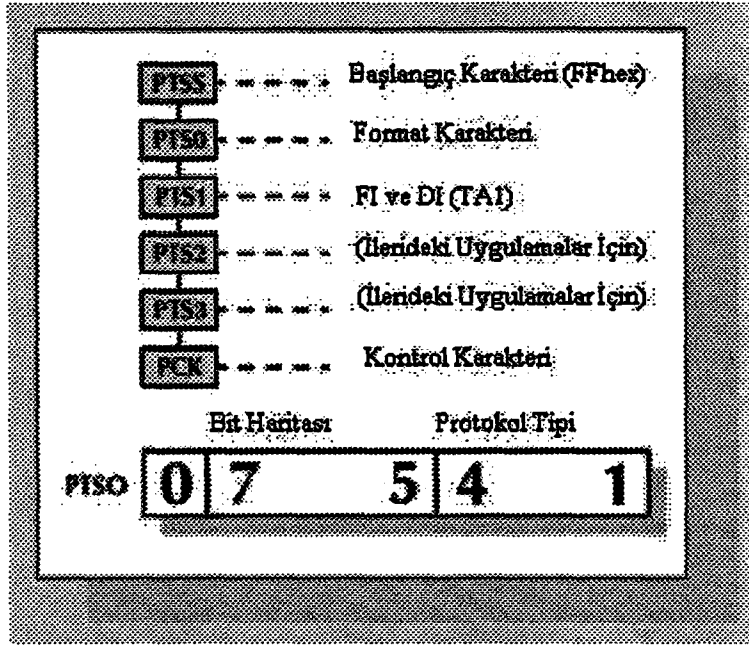
- Bir başlangıç karakteri PTSS (FFhex kodlanmış olarak),
- Bir format karakteri PTSO,
- Üç adet isteğe bağlı karakter PTS1, PTS2, PTS3,
- Kontrol karakteri PCK.

Bu yapı Şekil C.6'da gösterilmiştir. Kartın cevabı da istekle aynı yapıdadır.

PTSO format karakteri Şekil C.6'da görüldüğü gibi kodlanmıştır. Bit haritası PTS1, PTS2 ve PTS3'ün bulunup bulunmadığını göstermeye yarar. Bunlar sırası ile 5. , 6. ve 7. bitlerle kodlanırlar ve atanan mantıksal 1 değeri ilgili karakterin varlığına işaret eder. Protokol tipi 1., 2., 3. ve 4. bitlerde ikilik düzende T=0 ile T=15 değerleri arasında kodlanmıştır.

PTS1 karakteri eğer mevcutsa, FI için değerleri TA1 içinde kodlandığı şekilde tanımlamaya yarar. Kontrol karakteri PCK ise PTSO'dan PCK'ya kadar bütün bitlerin DARVEYA'lanması sonucu 0 olacak şekilde hesaplanır.

Akıllı kart, PTS istek mesajını doğru yorumlarsa, aynı mesajı geri göndererek cevap verir. PTS1 cevap karakterinin 5. biti 0'a çekildiğinde ise F ve D'nin varsayılan değerleri kullanılır.



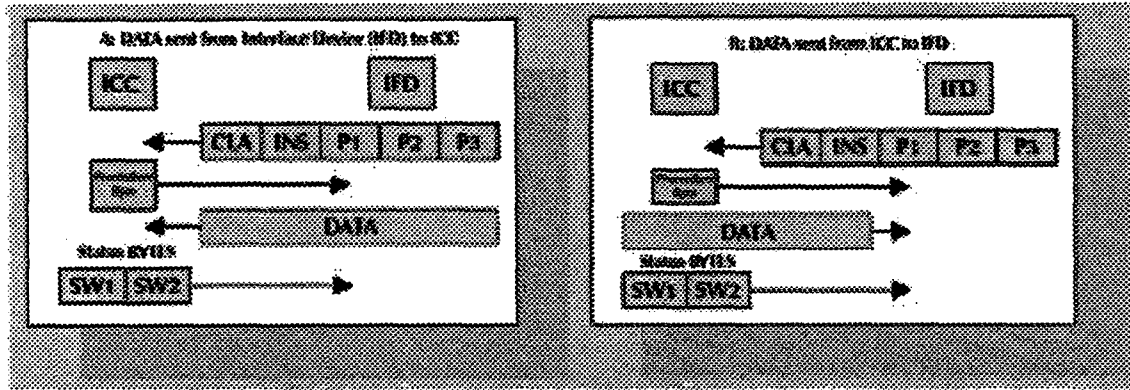
Şekil C.6. PTR İsteği

T=0 İletişim Protokolü

T=0 protokolü daima arabirim tarafından başlatılır. Arabirim ile kart arasındaki etkileşim, ardışıl komutlar ve cevaplar şeklinde sonuçlanır. Bu protokolda veri akışı sadece tek yönlü olur. Başka bir deyişle ya komut mesajı kart için veri taşır ya da karttan gelen komut isteği içinde cevap verisi barındırılır. Veri akışı yönü komutun tanımına bağlıdır ve bu sebepten arabirim ile kartın bir ön bilgiye ihtiyacı vardır. Özel bir komut için iki yönlü veri transferi gerekirse, ilk komuttan sonra bir “cevap al” komutu kullanılabilir. Arabirimin karta gönderdiği komut mesajı 5 karakterlik bir başlıktan oluşur. Kart da gelen komuta bağlı olarak bir prosedür byte’ı ardından veriyi arabirime gönderir veya arabirimden alır. Bu prosedür byte’ı, arabirimin Vpp EPROM programlama gerilimini kontrol etmesine izin vermek içindir. EEPROM bellek kullanıldığında bu prosedür byte’ı işe yaramaz ve gereksizdir. T=0 protokolü veri akışı Şekil C.7’de gösterilmiştir. Komut başlığı aşağıdaki 5 byte’tan oluşur.

- CLA : Komut sınıfı (FF değeri PTS için ayrılır)
- INS : Komut kodu (örneğin “bellek”)
- P1 : Komut kodu belirleyicisi (örneğin “bellek adresi”)
- P2 : Ek INS belirleyicisi
- P3 : Veri bloğu uzunluğu

P3 sıfır olduğunda karttan gelen veri 256 byte uzunluktadır. Eğer verinin karta aktarılması söz konusu ise, sıfır veri transferi uygulanır.



Şekil C.7. T=0 Protokolünde Veri Akışı

ACK prosedür byte'ı normalde komut byte'ını (INS) yankılar. Diğer seçenekler arabirimin Vpp programlama gerilimini gerektiği gibi kontrol etmesini sağlar. Kart NULL prosedür byte'ı (60hex) göndererek daha fazla işlem zamanı talep edebilir. Bu durumda IFD bir başka prosedür byte'ı bekler. ISO standardı aynı zamanda ilk durum byte'ının (SW1'in) prosedür byte'ı olarak gönderilmesine de izin verir. SW1 (status word) ve SW2 olmak üzere iki durum byte'ı vardır. Bu byte'lar kart tarafından komut sonlandığında mevcut kart durumunu belirtmek için arabirime gönderilir. Normal cevap:

SW1, SW2 = 90hex, 00hex

Eğer SW1 = 6X veya 9X ise, karttan çeşitli hata durumları rapor edilir. ISO 7816-3 bu tür 5 hata durumu tanımlar. Bu durumda SW1 aşağıdaki değerlerden birisini alır:

- 6E : Kart komut sınıfını desteklemiyor.
- 6D : Geçersiz INS (komut) kodu
- 6B : Yanlış Referans
- 67 : Yanlış uzunluk
- 6F : Belirsiz

T=1 İletişim Protokolü

T=1 iletişimi bir askenkron yarı çift yönlü protokoldür. OSI modelinde bu protokol 2. katmanda (veri bağı katmanında) çalışır. Fiziksel katman (1. katman) hata belirleme ve düzeltme özellikleri dışında T=0 protokolünde olduğu şekilde çalışır. Bu protokol bir karakter bloğunun çevresine bir zarf geçirerek aşağıdaki fonksiyonlara imkan sağlar.

- akış kontrolü
- blok zincirleme
- hata düzeltme

Kart için iletişim protokolü seçimi halen gündemde olan bir konudur ve blok protokolü ile sağlanan avantajlarla karşılığında yapılması gereken masraf iyi değerlendirilmelidir. T=1 protokolünün en dikkat çeken avantajı iki yönlü veri akışına imkan vermesidir. T=0 protokolündeki yön sınırlaması, komuta bağlı veri uzunluğunu belirlemek için tek bayt kullanılmasından kaynaklanıyordu.

T=1 protokolü aynı zamanda, T=0'daki daima arabirimin (IFD'nin) başlattığı, kartın cevap verdiği komut yapısındaki istemci-sunucu sınırlamasını da kaldırmaktadır. Bu blok protokolünde ise komut hem IFD hem de kart tarafından başlatılabilir. T=1'in diğer bir avantajı da veri bloklarını sıralı pencere numaraları ile zincirleyerek büyük veri gruplarını tek bir komutla iletme yeteneğidir.

Blok protokolü daha gelişmiş bir hata işleme sistemine sahiptir. Bu bir hata tanıma kodu (EDC-error detection code) kullanımına ve hatalı blokların tekrar gönderimine imkan verir. T=0 protokolünün daha basit bir hata tanıma ve düzeltme metodu vardır.

Bu yüksek katman protokolü için ödenmesi gereken bir bedel vardır. Kart ve IFD üzerinde koşan programların karmaşıklaşmasına ek olarak, son gönderilen bloğun muhtemel tekrarlar için tutulması da daha yüksek RAM bellek ihtiyacını doğurur. Genelde T=1 protokolü, uygulama büyük veri blokları kullandığında, özellikle komutlarda çift yönlü veri akışı gerektiğinde avantajlıdır. Protokolün etkinliği sadece gerçekten büyük veri transferlerinde ortaya çıkmaktadır, çünkü altta çalışmakta olan fiziksel katman halen T=0 protokolünün karakter kipini kullanmaktadır. Karakter çerçevesinin 11 etu'ya (elementary time units – temel zaman birimi) indirilmesi, T=0'da gereken 12 etu ile karşılaştırıldığında; hem başa hem de sona ekler getiren pencere yapısından kaynaklanan yönetimsel ekstra yükü dengelemek zorundadır.

Hata kontrolünün T=0 protokolüne nazaran iyileştirildiği şüphesizdir, ancak birçok kartın kısa mesafede 9600 bit/saniye hızında iletişim hataları zaten oldukça azdır. Günümüzde T=1 protokolüne eğilim artmaktadır ve gelecekte baskın protokol olacak gibi görünmektedir. Yine de bazı durumlarda daha verimli teknik çözüm sunabilen T=0 protokolünün kullanımı azımsanmamalıdır. T=1 protokolü ISO standardında ISO 7816-3 / AMD.1 olarak tanımlanmıştır.

Blok Penceresi

Pencere aşağıda görünen üç alandan oluşur.

- Prolog alanı (giriş)
- Bilgi alanı (seçime bağlı)
- Epilog alanı (sonuç)

Prolog			Bilgi	Epilog
NAD	PCB	LEN	INF	EDC
1	1	1	0-254	1 / 2

Şekil C.8. Blok Penceresi

Prolog alanı üç byte içerir.

- NAD (node address) adres
- PCB (protocol control byte) protokol kontrol byte'ı
- LEN (length) veri uzunluğu

NAD byte'ının 1-3 numaralı bitlerini kaynak adresini, 5-7 numaralı bitleri ise hedef adresini belirler. 4. ve 8. bitler ise Vpp kontrolü için kullanılır. NAD gerektiğinde birden fazla mantıksal kanal kullanımına izin verir; aksi taktirde iki adres de 0'a çekilmelidir.

PCB byte'ı üç tip blok penceresi tanımlamaya yarar:

- I - Bilgi Bloğu
- R - Almaya Hazır Bloğu
- S – Danışma Bloğu

Bilgi bloğu, uygulama komutlarını ve kart ile IFD arasında taşınan veriyi göndermeye yarayan penceredir. Almaya hazır bloğu ise, protokol zincirli bloklar dizisi halinde veri gönderirken onay (ACK) olarak kullanılır. Danışma bloğu kontrol parametrelerini ve hata sonuçlarına bağlı olarak tekrar senkronizasyon ve başarısız çıkış etkilerini ayarlamaya yarar. Bilgi bloğu aynı zamanda zincirsiz kipte onay (ACK) byte'ı olarak da davranır.

LEN byte'ı pencerenin bilgi alanındaki byte sayısını belirtir. İzin verilen değerleri 00-FEhex arasındır. Bu en fazla 254 byte'lık bir bilgi alanına izin verir. Bilgi alanı uygulama komutlarını ve veriyi tutmaya yarar.

Epilog alanı blok hata yakalama kodunu içerir. Bu kod LRC (longitudinal redundancy check) veya CRC (cyclic redundancy check) olabilir. LRC 1 byte, CRC ise 2 byte'tır. Bu seçenek özel arayüz karakterleri ile tanımlanır.

Özel Arayüz Karakterleri

T=1 protokolü bu karakterlerden 3 tanesini iletişim başlamadan önce gerekli seçenekleri ayarlamak için kullanır. Bu byte'lara değerler aşağıdaki şekilde atanır. (i>2 için)

- $TA_i = IFSC$ (varsayılan = 32)
- $TB_i = \text{bit 1-4} = CWI$ (varsayılan = 13)
bit 5-8 = BWI (varsayılan = 4)
- $TC_i = (\text{bit 1} = 1) \rightarrow \text{CRC seçeneği}$
(bit 1 = 0) $\rightarrow$ LRC seçeneği (varsayılan)

IFSC kart için bilgi alanı uzunluğudur. IFSD ise arabirimin veri alanı uzunluğudur. Bunun varsayılan değeri 32 byte'tır ve sadece IFD veya kart tarafından bir S- Blok isteği geldiği zaman değiştirilir.

Bekleme Süreleri

T=1 protokolü veri akışına yardımcı olması için iki bekleme süresi parametresi kullanır.

- Karakter Bekleme Süresi (KBS)
- Blok Bekleme Süresi (BBS)

KBS bir bloktaki başarılı karakterlerin arasındaki maksimum süredir. BBS ise IFD'den gönderilen bir bloğun başlangıç zamanı ile karttan gönderilen bir sonraki bloğun başlangıç zamanı arasındaki farktır. KBS blok uzunluğundaki bir hatayı yakalamayı sağlayabilirken, BBS de karşılık vermeyen bir kartı farketmeye yarar. Blok koruma zamanı (BKZ) ise, giden ve gelen blokların başlangıçları arasındaki minimum zaman dilimidir. KBS ve BBS yukarıda anlatılmış olan CWI ve BWI'nın kodlanmış değerleri ile hesaplanır.

$$KBS = (2 BWI + 11) \text{ etu} \quad (2.2)$$

$$BBS = (2 BWI * 960 * 372 / f) \text{ saniye} + 11 \text{ etu} \quad (2.3)$$

En düşük BWI değeri 100 ms+11 etu 'dur, eğer kart varsayılan frekansı olan 3.58 MHz ile çalışırsa. BKZ = 22 etu'dur; alınan bir blok ile karşılığında gönderilen bloğun arasındaki gecikme BKZ'den büyük ve BBS'den küçük olacak şekilde. Buna bağlı olarak bloklar arası minimum süre bir karakter süresine karşılık düşen 11 etu'dur.

Protokol Kontrol Byte'ı (PKB)

Protokol kontrol byte'ı değişik blok tiplerini tanımlar ve bazı kontrol bilgilerini taşır. Bu kontrol bilgileri, 1 bitlik bir sıra numarasını (N) ve bir blok zincirleme bitini (M) içerir. Diğer bitler iletişim hatalarını tanımlamak için kullanılır. PKB Tablo C.1'de gösterildiği gibi kodlanır.

Tablo C.1. PKB Kodlama Tablosu

Tip	PKB (1-8. bitler)								Fonksiyon
I	0	N	0	0	0	0	0	0	Standart I Bloğu
I	0	N	1	0	0	0	0	0	Zincir bit kümesi
R	1	0	0	N	0	0	0	0	Hata yok
R	1	0	0	N	0	0	0	1	EDC / Parite hatası
R	1	0	0	N	0	0	1	0	Diğer hatalar
S	1	1	0	0	0	0	0	0	Tekrar senkronizasyon isteği
S	1	1	1	0	0	0	0	0	Tekrar senkronizasyon cevabı
S	1	1	0	0	0	0	0	1	IFS isteği
S	1	1	1	0	0	0	0	1	IFS cevabı
S	1	1	0	0	0	0	1	0	Çıkma isteği
S	1	1	1	0	0	0	1	0	Çıkma cevabı
S	1	1	0	0	0	0	1	1	WTX isteği
S	1	1	1	0	0	0	1	1	WTX cevabı

I blokları bağımsız ya da bir zincirin parçası olabilirler. "Daha fazla" biti, takip eden bloklar olduğunu belirtir. Gönderenin sıra numarası 0 ile başlar ve sürekli 0 ile 1 arasında salınır.

R blokları iletilen blokların, başarılı ya da başarısız olması durumunu onaylamak için kullanılır. Sıra numarası N beklenen bir sonraki bloktaki N değerini taşır. Zincirleme gönderilen I bloklarının her birine R ile cevap vermek gerekirken, yalnız başına gönderilen bir I bloğuna yine bir I bloğu ile cevap verilebilir. Karşılıklı olarak sıra numaraları sürekli 0 ile 1 arasında geçiş yapar. Tabloda görüldüğü gibi R bloğunun 3 muhtemel durumu vardır.

S blokları tabloda görüldüğü gibi 4 kontrol durmunu uyandırmaya yarar. Senkronizasyon isteği iletişim parametrelerini başlangıç değerlerine almak için sadece IFD tarafından kullanılır. Zincirleme iletişim ise bir fiziksel hata veya bellek arızası durumunda hem IFD hem de kart tarafından terkedilebilir. Kart, bir IFS isteği göndererek IFSC'de desteklenen bir değişikliği tetikleyebilir. Benzer şekilde IFD de bir IFS isteği göndererek IFSD değişikliği talep edebilir. S blok kontrolü ayrıca karta BWT'ye bir ek yapma imkanı verir; bu da bir I bloğu içinde alınan bir komutu

çalıştırmak için gerekebilir. Bu bloktaki INF alanı BBS değerinin tam katı olarak hesaplanabilen tek byte'lık bir sayısal değer içerir. Her durumda bir S bloğu alan, uygun yanıt bloğunu göndermelidir.

İşlem / Operasyon Sırasında T=1 Protokolü

ISO 7816 standardının yazım şekli kullanılarak protokolün basit işleme şekli gösterilebilir. Daha kapsamlı bir tanım ise standarttan elde edilebilir.

- I Blokları : I (N,M)

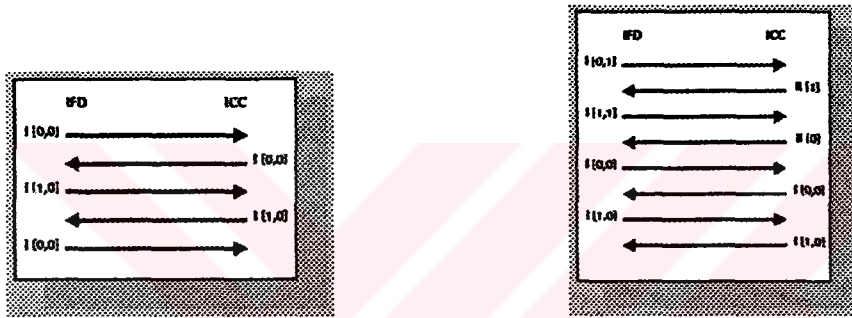
N = Sıra numarası (0/1)

M = Daha fazla veri biti (eğer takip eden I bloğu varsa 1'e çekilir)

- R Blokları : R(N)

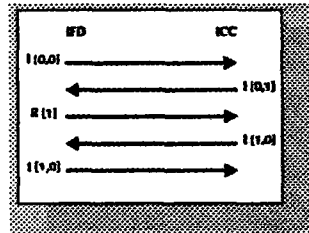
N = Beklenen bloğun sıra numarası

Protokol IFD ve kartın her ikisine de sıra ile gönderme hakkı verir ve iletişim IFD'nin bir bloğu ile başlar ve Şekil C.9'da gösterilmiştir.



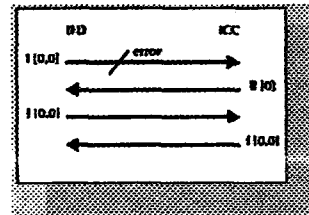
Şekil C.9. T=1 Protokolü ve Normal I İletişimi Gönderimi

Kartın IFD tarafından gönderilen zincirin son bloğuna onay için I bloğu kullandığına dikkat edilmelidir. Şekil C.10'da gösterildiği gibi kart da aynı yöntemle zincirleme bloklar gönderebilir.



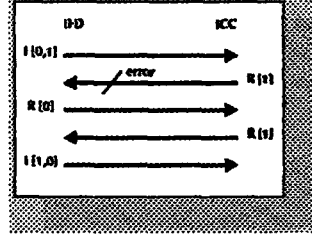
Şekil C.10. I Blok İletişiminde Zincirleme

Şekil C.11'de I Blok gönderiminde hatanın nasıl koterıldığı gösterilmiştir.



Şekil C.11. I blok alımında hata

I bloğu zincir cevabında hata ise Şekil C.12’de verilmiştir.



Şekil C.12. I blok Yanıt Zincirinde Hata

Her iki durumda da gönderici hatalı ulaşan bloğu tekrar göndermek üzere bilgilendirilir. Elbette ki birçok hata ihtimali senaryosu mevcuttur, ancak hepsi yukarıda anlatılana benzer şekilde ele alınır.

Akıllı kartların kullanımına geçilmesindeki yavaşlık çok sorgulanmış ve buna verilen tipik cevap standartların bulunmaması olmuştur. Burada standartların mevcut olduğu gösterilmiştir.

ISO standartlarına uygun olarak akıllı kart kullanan bir çok ticari proje bulunmaktadır. (Mondex gibi - Küresel Elektronik Nakit Para) Benzer şekilde Visa, Mastercard ve Europay de değişik ödeme sistemlerine ISO standardı üzerine kurulu özelliklerde kırmık kartları üretmek için birlikte çalışmışlardır.

ÖZGEÇMİŞ

Ece ÇİNER, 1975 yılında İzmir’de doğmuştur. İlk ve orta öğrenimini İzmir’de tamamlayan ÇİNER, 1993 yılında Özel İzmir Amerikan Lisesi’ni bitirmiştir. 1997 Yılında İstanbul Teknik Üniversitesi Kontrol ve Bilgisayar Mühendisliği Lisans programından mezun olmuş, ardından İTÜ’de Kontrol ve Bilgisayar Mühendisliği Ana Bilim Dalı’nda yüksek lisansa başlamıştır. 1997’den bu yana Bilgi Teknolojileri ve Finans sektörlerinde hizmet veren çeşitli şirketlerde bankacılık, elektronik ticaret ve alternatif dağıtım kanalları ile ilgili projelerde görev almıştır. Halen özel bir kuruluştaki kart yönetimi ve ödeme sistemleri konusunda yazılım geliştirme uzmanı olarak çalışmaktadır.

