



Advanced RFID Card Reader Writer

RFIDAX Cihaz Serisi

USB NFC Reader Writer

Reference Manual V2.00



Advanced RFID Card Reader Writer

Revizyon Geçmişi

Tarih	Değişiklikler	Versiyon
2025-02-05	İlk Sürüm	0.01
2025-04-24	Rfid kart format özelliği eklendi	0.02



Advanced RFID Card Reader Writer

İçindekiler

Şekil Listesi	9
Tablo Listesi	9
1.0 Giriş	11
2.0 Özellikler	12
3.0 RFIDAX Serisi Mimarisi	13
3.1 Cihaz Okuyucu Yazıcı Blok Diyagramı	13
3.2 İletişim Protokolü Yapısının Blok Diyagramı	14
4.0 RFIDAX Cihazlarının Model ve Cihaz numaraları	15
5.0 Donanım Tasarımı	16
5.1 USB Bağlantı	16
5.1.1. İletişim Parametreleri	16
5.2 RS485 Bağlantı	16
5.2.1. İletişim Parametreleri	16
5.3 Type-C Bağlantı	17
5.3.1. İletişim Parametreleri	17
5.4 Temassız Akıllı Kart Arayüzü	18
5.4.1. Taşıyıcı Frekansı	18
5.4.2. Kart Tarama (Polling)	18
5.5 Kullanıcı Arayüzü	19
5.5.1. Buzzer ve LED	19
6.0 Cihaz Yazılımsal Komutları ve Konfigasyonu	20
6.1 Cihazın Modları ve Açıklamaları	20
6.2 Veri Bütünlüğü Modları ve Açıklamaları	21
6.2.1 Veri bütünlüğü Pasif Olduğu Durum	21
6.2.2 Veri bütünlüğü Aktif Olduğu Durum	21
6.3 Cihazların Haberleşme Türleri	22
6.4 Cihaz Durum Mesajı, Hata Kodları ve Açıklamaları	22
6.4.1 Genel Sistem Durumu ve Hataları	22
6.4.1.1 Genel Sistem Hataları Ek Açıklama	22
6.4.2 Zaman Aşımı ve Veri Hataları	23
6.4.2.1 Zaman Aşımı ve Veri Hataları Ek Açıklama	23
6.4.3 LED ve Göstergeler ile İlgili Hatalar	23
6.4.3.1 LED ve Göstergeler ile İlgili Hataların Ek Açıklaması	23
6.4.4 RFID Kart Okuma/Yazma Hataları	24
6.4.4.1 RFID Kart Okuma/Yazma Hatalarının Ek Açıklaması	25



Advanced RFID Card Reader Writer

6.4.5 İleri Seviye Hatalar ve Sistem Yönetimi	26
6.4.5.1 İleri Seviye Hatalar ve Sistem Yönetimi Hatalarının Ek Açıklamaları	27
6.5 Cihaz Sistem İzleme Döngüsü (System Monitor Loop)	28
6.5.1 Sistem İzleme Döngüsünün Çalışma Prensipleri	28
6.5.2 Kart Algılama ve Karşılaştırma Süreci	28
6.5.2.1 LED ve Buzzer Bildirimleri	29
6.5.3 Kart Kaldırıldığında Yapılan İşlemler	29
6.5.4 Flash Bellek ile Çalışma	29
6.5.5 Sistem İzleme Döngüsünün Önemi	29
6.6 Kart okutulduğunda Cihazdan Otomatik Gelen Cevap	30
6.6.1 Tanımlar ve Veri Yapısı	30
6.6.2 Byte Byte Açıklama	32
6.6.2.1 Byte Byte Açıklama Özeti	33
6.6.2.1.1 Örnek Sistem Monitor Mesajı Açıklaması	34
6.7 Cihazdan Gelen mesaj yapısı	37
6.7.1 Durum Mesajları (Report_status)	37
Durum Mesajları (Report_status) Örneği 1	38
Durum Mesajları (Report_status) Örneği 2	38
Durum Mesajları (Report_status) Örneği 3	39
Durum Mesajları (Report_status) Örneği 4	39
Durum Mesajları (Report_status) Örneği 5	40
6.7.2 Veri Paketleri (send_data_packet)	41
6.7.2.1 Message Type Bilgisi	42
6.7.2.1.1 RFID Kart Okuma İşlemleri Mesajları ve Message Type Bilgisi	42
6.7.2.1.2 Flash Bellek ID Güncelleme Mesajları ve Message Type Bilgisi	43
6.7.2.1.3 Flash Bellekten ID Okuma Mesajları ve Message Type Bilgisi	43
6.7.2.1.4 Flash Bellek Sıfırlama Mesajları ve Message Type Bilgisi	44
6.7.2.1.5 Format ID Okuma Mesajı ve Message Type Bilgisi	44
6.7.2.1.6 Sector Trailer İşlemleri ve Message Type Bilgisi	44
Veri Paketleri (send_data_packet) Örneği 1	45
Veri Paketleri (send_data_packet) Örneği 2	45
Veri Paketleri (send_data_packet) Örneği 3	46
Veri Paketleri (send_data_packet) Örneği 4	46
7.0 Cihaz Komutları Kılavuzu	47
7.1 Komut Kategorileri ve Açıklamaları	47
7.1.1 Komut Kategorileri ve Alt Kategorilerin Çalışma Modlarına Açıklamaları	48
7.2 LED ve Buzzer Kontrol Komutları	50
7.2.1 Cihazdaki LEDlerin Kontrol Komutları	50



Advanced RFID Card Reader Writer

7.2.1.1 Cihazdaki Ledlerin Led Aç/Kapat Komutları.....	50
7.2.1.1.1 Mavi veya Sarı Ledin Kontrol Komutları	50
7.2.1.1.2 Kırmızı Ledin Kontrol Komutları	51
7.2.2 Cihazdaki Ledlerin LED Toggle Komutları.....	51
7.2.2.1 Mavi veya Sarı Ledin Toggle Komutları.....	52
7.2.2.2 Kırmızı Ledin Toggle Komutları.....	52
7.2.3 Cihazdaki Ledlerin Çif LED Toggle Komutları	53
7.2.3.1 Cihazdaki Ledlerin Çif LED Toggle Başlatma	53
7.2.3.2 Cihazdaki Ledlerin Çif LED Toggle Durdurma	54
7.2.4 Cihazdaki Tüm Ledleri Kapatma Komutları	54
7.2.5 Cihazdaki Buzzer Aç/Kapatma Komutları	55
7.2.5.1 Buzzer Aç/Kapat Komutu.....	55
7.2.6 Cihazdaki Süreli Buzzer Aç/Kapatma Komutları.....	56
7.2.6.1 Buzzer Süreli Aç/Kapat Komutu	56
7.2.7 Led ve Programlabılır Buzzer Tüm Komutları	57
7.3 RFID Kart İşlem Komutları.....	58
7.3.1 RFID Kart Tanıma ve Okuma/Yazma Komutları	58
7.3.2 RFID Kart Tanıma Komutları.....	59
7.3.2.1 RFID Kart UID Okuma Komutu	59
7.3.2.2 RFID Kart SAK Değeri Okuma Komutu	60
7.3.2.3 RFID Kart ATQA Değeri Okuma Komutu	61
7.3.2.4 Kart bilgilerini okuma (UID + SAK + ATQA) Komutu	62
7.3.2.5 RFID Kart Anahtarlarını Gösterme Komutu	63
7.3.2.5.1 Sabit Anahtarlar (KeyA ve KeyB)	64
KeyA Okuma.....	64
KeyB Okuma	64
7.3.2.5.2 Kullanıcı Tarafından Tanımlanabilen Anahtarlar (Optional KeyA ve Optional KeyB)	65
Optional KeyA Okuma.....	65
Optional KeyB Okuma.....	65
7.3.2.5.3 Tüm Anahtarları Okuma (KeyA, KeyB, Optional KeyA ve Optional KeyB).....	66
7.3.3 RFID Kart İçindeki Blokların İçine Veri Yazma ve Bloktan Veri Okuma İşlemi.....	67
7.3.3.1 RFID Kart Bloklarına Veri Yazma Komutu.....	67
7.3.3.1.1 RFID Kart Bloklarına Veri Yazma Komutu Örneği	69
7.3.3.1.2 RFID Kart Bloklarına Veri Yazma Komutu Örneği -2.....	71
7.3.3.1.3 RFID Kart Bloklarına Veri Yazma Komutu Örneği -3.....	73
7.3.3.2 RFID Kart Bloklarına Veri Okuma Komutu	75
7.3.3.2.1 RFID Kart Bloklarından Veri Okuma Komutu Örneği-1	77
7.3.3.2.2 RFID Kart Bloklarından Veri Okuma Komutu Örneği-2	78



Advanced RFID Card Reader Writer

7.3.3.2.3 RFID Kart Bloklarından Veri Okuma Komutu Örneği- 3	80
7.4 Sistem Kontrol Komutları	81
7.4.1 Sistem Kontrol Komutlarının Kullanım Amaçları	81
7.4.2 Sistem Kontrol Komutları ve Listesi	81
7.4.2.1 Yazılım Sıfırlama Komutu	82
7.4.2.2 Cihaz Versiyon Bilgisi Alma Komutu	83
7.4.2.2.1 Cihaz Versiyon Bilgisi Alma Komutu Genel Mesaj Yapısı	83
7.4.2.2.1.1 Cihaz Versiyon Bilgisi Alma Komutu Genel Mesaj Yapısı Örneği 1	84
7.5 Bellek Yönetim Komutları	86
7.5.1 Bellek Yönetim Komutlarının Kullanım Amaçları	86
7.5.2 OptionalKeyA ve OptionalKeyB İşlemleri	87
7.5.2.1 OptionalKeyA ve OptionalKeyB Komutlarının Kullanım Amaçları	87
7.5.2.2 Gönderilen Mesaj Formatı	87
7.5.2.3 Gelen Mesaj Formatı (Yazma İşlemi Sonrası)	87
7.5.2.4 Gelen Mesaj Formatı (Okuma İşlemi Sonrası)	88
7.5.2.5 OptionalKeyA Yazma Komutu Örneği	88
7.5.2.6 OptionalKeyA Okuma Komutu Örneği	89
7.5.2.7 OptionalKeyB Yazma Komutu Örneği	90
7.5.2.8 OptionalKeyB Okuma Komutu Örneği	91
7.5.2.9 Özet	91
7.5.2 ID İşlemleri	92
7.5.3.1 ID İşlemlerinin Kullanım Amaçları	92
7.5.3.2 Bellek Yönetimi ve ID İşlemlerinin Önemi	92
7.5.3.3 Gönderilen Mesaj Formatı	93
7.5.3.4 Gelen Mesaj Formatı (ID Güncelleme, Okuma ve Sıfırlama Yanıtı)	93
7.5.3.5 ID Güncelleme Komutu Örneği	94
1. Örnek: ID 0 Güncelleme İşlemi	94
2. Örnek: ID 9 Güncelleme İşlemi	95
7.5.3.6 ID Okuma Komutu Örneği	96
1. Örnek: ID 1 Okuma İşlemi	96
2. Örnek: ID 9 Okuma İşlemi	97
7.5.3.7 ID Sıfırlama Komutu Örneği	98
1. Örnek: ID 1 Sıfırlama İşlemi	98
2. Örnek: ID 9 Sıfırlama İşlemi	99
7.5.3.7 Tüm ID'leri Listeleme	100
7.6 Cihaz Yapılandırma Komutları	101
7.6.1 Cihaz Yapılandırma Komutlarının Kullanım Amaçları	101
7.6.2 Cihaz Yapılandırma Komutları Listesi	101

7.6.3 Device ID Deęiřtirme Komutu.....	102
7.6.3.1 Kullanım Amaçları	102
7.6.3.2 Gönderilen Mesaj Formatı	102
7.6.3.3 Gelen Mesaj Formatı (Device ID Deęiřtirme Yanıtı)	102
1. Örnek: Device ID 3'e Güncelleme	103
7.6.4 Protokol Deęiřtirme Komutu	104
1.Örnek Protokol Güncelleme İşlemi	106
7.6.6 Baud Rate Güncelleme İşlemi.....	107
7.6.6.1 Desteklenen Baud Rate Deęerleri	107
7.6.6.2 Gönderilen Mesaj Formatı	107
1 Örnek: Baud Rate Güncelleme İşlemi	109
7.7 Fabrika Ayarlarına Dönüş Komutları	110
7.7.1 Tüm Ayarları Fabrika Varsayılanlarına Sıfırlama Komutu	111
1.Örnek: Fabrika Ayarlarına Sıfırlama İşlemi.....	112
7.7.2 Format RFID Kartının ID'sini Okuma Komutu.....	113
1. Örnek: Format RFID Kartının ID'sini Okuma	114
7.7.3 Format Bayraęını Set/Reset Etme Komutu	115
7.7.3.1 Format Bayraęını Reset Etme (0x00 - Pasif Hale Getirme)	116
7.7.3.2 Format Bayraęını Set Etme (0x01 - Aktif Hale Getirme)	117
7.7.4 Manuel Olarak Format ID'yi Flash Belleęe Yazma.....	118
1.Örnek: Manuel Olarak Format ID'yi Flash Belleęe Yazma İşlemi.....	119
7.8 Sector Trailer Yönetim Komutları.....	120
7.8.1 Sector Trailer Okuma Komutu	121
1. Örnek: Sector Trailer Veri Yapısı Oluřturma İşlemi	122
7.8.2 Yalnızca Sector Trailer Bloklarını Özel Key'lerle Güncelleme	123
1. Örnek: Yalnızca Sector Trailer Bloklarını Güncelleme İşlemi	124
7.8.3 Tüm Veri Bloklarını Sıfırlama ve Sector Trailer Yapılandırması	125
1. Örnek: Tüm Veri Bloklarını Sıfırlama ve Sector Trailer Yapılandırma İşlemi	126
7.8.4 Kartı Tamamen Formatlama ve Fabrika Ayarlarına Döndürme.....	127
1. Örnek: Kartı Tamamen Formatlama ve Fabrika Ayarlarına Döndürme İşlemi.....	129



Advanced RFID Card Reader Writer



Şekil Listesi

Şekil 1 RFIDAX Okuyucu Blok Diyagramı	13
Şekil 2 RFIDAX Cihazları Haberleşme Blok Diyagramı	14
Şekil 3 Reset sonrası ASCII mesajı (Örnektir)	82

Tablo Listesi

Tablo 1 Cihaz Model numaraları ve model bilgisi	15
Tablo 2 USB Arayüzü Kablolama	16
Tablo 3 RS485 Arayüzü Kablolama	16
Tablo 4 TYPE-C Arayüzü Kablolama	17
Tablo 5 BUzzer ve LED Göstergeleri	19
Tablo 6 Cihaz Modları ve açıklamaları	20
Tablo 7 Veri Bütünlüğü Modları ve Veri Bütünlüğü Modları Açıklamaları	21
Tablo 8 CRC modları ve modların desteklediği alt CRC modları	21
Tablo 9 Haberleşme Türleri ve Haberleşme Türleri Açıklamaları	22
Tablo 10 Genel Sistem Hataları ve Açıklamaları	22
Tablo 11 Zaman Aşımı ve Veri Hataları ve Açıklamaları	23
Tablo 12 LED ve Göstergeler ile ilgili Hatalar ve Açıklamaları	23
Tablo 13 RFID Kart Okuma/Yazma Hataları ve Açıklamaları	24
Tablo 14 İleri Seviye Hatalar ve Sistem Yönetimi ve Açıklamaları	26
Tablo 15 LED ve Buzzer Bildirimleri	29
Tablo 16 Haberleşme Türleri	31
Tablo 17 Veri bütünlüğü modları	31
Tablo 18 Çalışma (operasyon Modları) Modları	31
Tablo 19 Sistem monitoring yapısının Byte Byte Açıklama	32
Tablo 20 Sistem monitoring yapısının Byte Byte Açıklama Özeti	33
Tablo 21 Sistem monitoring yapısı için örnek (Gerçek RFIDAX cihazı ve gerçek karttan veri alınmıştır)	34
Tablo 22 Sistem monitoring yapısı için örnek (Gerçek RFIDAX cihazı ve gerçek karttan veri alınmıştır)	35
Tablo 23 Sistem monitoring yapısı için örnek (Gerçek RFIDAX cihazı ve gerçek karttan veri alınmıştır)	36
Tablo 24 (Gerçek RFIDAX cihazı ve gerçek karttan veri alınmıştır)	38
Tablo 25 Message Type Bilgisi	42
Tablo 26 RFID Kart Okuma İşlemleri Mesajları ve Message Type Bilgisi	42
Tablo 27 Flash Bellek Güncelleme Mesajları ve Message Type Bilgisi	43
Tablo 28 Flash Bellekten Okuma Mesajları ve Message Type Bilgisi	43
Tablo 29 Flash Bellek Sıfırlama Mesajları ve Message Type Bilgisi	44
Tablo 30 Flash Bellek Sıfırlama Mesajları ve Message Type Bilgisi	44
Tablo 31-A Sector Trailer Formatlama Mesajı ve Message Type Bilgisi	44
Tablo 32 Komut Bileşenleri ve Açıklamaları	47



Advanced RFID Card Reader Writer

Tablo 33 Komut KategorileriVe Alt Katagorilerin Çalışma Modlara Açıklamaları	49
Tablo 34 Buzzer Süreli Aç/Kapat Saniye Hesaplama Örneği	56
Tablo 35 LED ve Buzzer Kontrol Komutları Tablosu	57
Tablo 36 Sistem kontrol komutları	81
Tablo 37 Bellek Yönetim Komutları.....	86
Tablo 38 OptionalKeyA ve optionalKeyB İşlemleri	86
Tablo 39 ID İşlemleri	86
Tablo 40 Cihaz Yapılandırma Komutları Listesi.....	101
Tablo 41 Desteklenen Protokoller	104
Tablo 42 Desteklenen Baud Rate Değerleri	107
Tablo 43 Fabrika Ayarlarına Dönüş Komutları Listesi.....	110
Tablo 44 Format Bayrağını Set (aktif) veya Reset (pasif) Durumu	117
Tablo 45 Sector Trailer Yönetim Komutları Listesi	120
Tablo 46 Format İşlem Sonucunda Kartın Yapısı.....	127



Advanced RFID Card Reader Writer

1.0 Giriş

RS485-RW Series TC-RW Series TC-RD Series MUSB-RD Series MUSB-RW Series HID-TC Series HID- MUSB Series
RFIDAX tarafından geliştirilen RFID okuyucu ve yazıcı cihazlar, 13,56 MHz temassız teknolojiye dayalı olup, entegre NFC işlevselliği ile üstün özelliklere sahiptir.

RFIDAX tarafından geliştirilen cihaz, ISO/IEC 14443 Type A protokolü ve MIFARE ailesine ait protokolleri tam olarak desteklemekte, NFC teknolojisiyle uyumlu çalışmaktadır.

RFIDAX tarafından geliştirilen cihazlar, 13.56 Mhz bandında çalışan beşinci nesil cihazlardır.

Bu cihazlar üst teknoloji RFID okuma ve yazma işlemleri gerçekleştirir. RFIDAX tarafından tasarlanan 13.56 Mhz frekans bandında çalışan cihazların tamamı temassız bir okuyucu ve yazıcıdır.

HID desteğiyle donatılmış sistem, USB üzerinden hızlı ve sürücüsüz iletişim sağlayarak tak ve çalıştır özelliğiyle kolay bir entegrasyon sunar. Bunun yanı sıra, standart USB protokolü sayesinde yüksek hızlı veri aktarımı desteklenir, bu da farklı iletişim gereksinimlerine uygun esnek bir altyapı sağlar. HID ve standart USB protokollerinin birbirinden bağımsız olarak çalışabilmesi, farklı uygulama senaryoları için optimize edilmiş bir çözüm sunar.

Sistem ayrıca, RS485 fiziksel katmanı üzerinden güvenilir ve uzun mesafeli veri aktarımı gerçekleştirebilir. Diferansiyel sinyal yapısına sahip RS485, parazitli ortamlarda bile kararlı ve güvenilir iletişim sağlarken, USB arayüzü hızlı veri aktarımı ve aygıt yazılımı güncelleme desteği sunar.

Endüstriyel otomasyon, bina yönetim sistemleri, enerji izleme ve kontrol, akıllı şehir altyapıları, erişim kontrol sistemleri ve finansal veri işleme gibi geniş bir uygulama yelpazesiyle uyumlu olan sistem, Modbus protokolü ve CRC tabanlı özel bir iletişim yapısı ile özelleştirilebilir veri paketleri ve esnek bir komut mimarisi sunar. Bu çok yönlü iletişim altyapısı, ticari ve endüstriyel uygulamalarda güçlü, güvenilir ve esnek bir çözüm sağlar.

Cihaz, USB, RS485, Type-C ve mobil iletişim gibi çeşitli haberleşme arayüzlerini destekleyerek çok yönlü ve esnek bir mimari sunar. Değiştirilebilir adres, ayarlanabilir haberleşme hızı ve esnek haberleşme protokolü ile yüksek uyumluluk ve kullanım kolaylığı sağlar. Hızlı önceki ayarlara dönme modu sayesinde, her koşulda güvenilir ve kararlı bir çalışma ortamı sunar. Veri bütünlüğü için CRC aktif veya pasif cihazları arasında seçim yapılmasına olanak tanır ve Temel, Gelişmiş, Kurumsal ve Özelleştirilmiş gibi farklı operasyon modlarını destekleyerek çeşitli uygulama gereksinimlerine uyum sağlar.

Ayrıca cihaz, iki NFC moduna akıllı destek sunar: okuyucu/yazıcı ve klavye emülasyonu. Esnek tasarımı sayesinde, e-ödeme sistemleri, erişim kontrolü ve ulaşım biletleme gibi geniş bir uygulama yelpazesinde sorunsuz bir şekilde entegre edilebilir. Gelişmiş yetenekleri, cihazı ticari ve endüstriyel kullanım senaryoları için güvenilir, güçlü ve ayarlanabilir bir çözüm haline getirir.

RFIDAX cihazları, 13.56 MHz kart okuyucu ve yazıcı teknolojisinde rakip ürünlere kıyasla üstün haberleşme hızı, farklı haberleşme protokollerini destekleme yeteneği, çeşitli fiziksel topolojilere uyumluluğu ve gelişmiş teknoloji altyapısı ile sektörde öne çıkmaktadır.

Gelişmiş teknolojisi sayesinde pazarda rakiplerinden ayrılarak lider konumda yer almaktadır.



Advanced RFID Card Reader Writer

2.0 Özellikler

✓ Performans

- ❖ RS485 Haberleşme Hızı: 256 kbps'ye kadar (istenirse hızlandırılabilir)
- ❖ USB Haberleşme Hızı: 12000 kbps'ye kadar
- ❖ ISO 14443A Okuma/Yazma Hızı: 848 kbps'ye kadar (istenirse hızlandırılabilir)
- ❖ Çalışma anında cihaz adresi, haberleşme hızı değişikliği ve haberleşme protokolü değişikliği
- ❖ Hızlı ve güvenilir şekilde çalışma anında fabrika ayarlarına hızlı bir şekilde geri dönüş (savunma sanayi, uzay sanayi)
- ❖ Programlanabilir led ve buzzer yapısı

✓ Çeşitli temassız kart türlerini destekler

- ❖ ISO 14443 Type A (Parts 1-4)
- ❖ MIFARE® (T=CL)
- ❖ MIFARE Classic
- ❖ MIFARE Ultralight
- ❖ MIFARE DESFire EV1
- ❖ MIFARE Plus
- ❖ MIFARE Mini
- ❖ NFC Tags

✓ Çeşitli NFC özelliklerini destekler

- ❖ Kart Okuyucu/Yazıcı Modu
- ❖ Sadece Kart okuyucu Modu
- ❖ Klavye Modu

✓ Güvenlik

- ❖ **Gelişmiş Yazılım Kriptografisi:** Üst düzey güvenlik için yazılım tabanlı kriptografi desteği.
- ❖ **Ücretsiz USB Yazılım Güncellemesi:** En son özellikler ve iyileştirmeler için kolay güncelleme imkanı

✓ Güvenilirlik ve Dayanıklılık

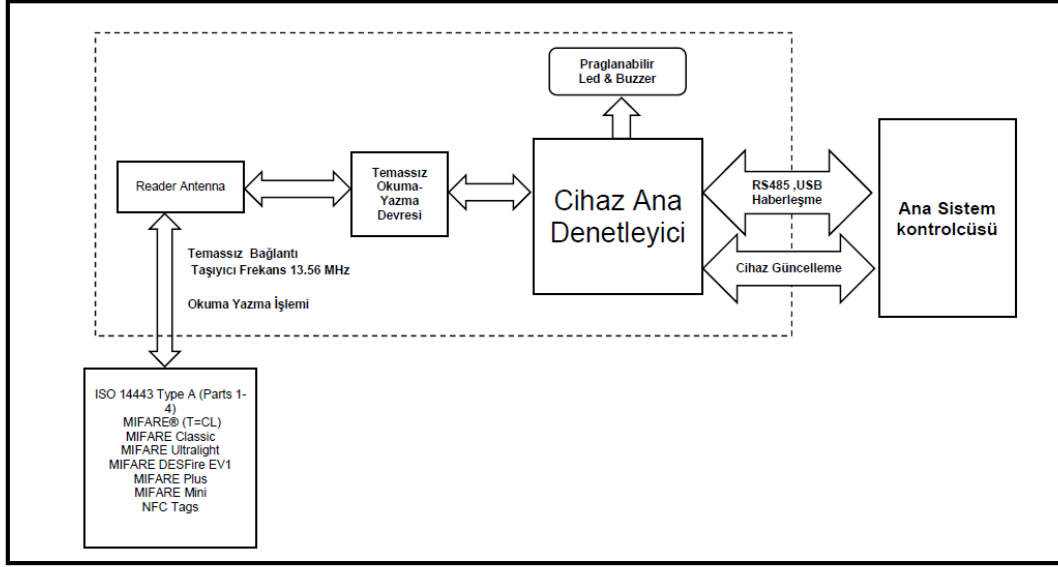
- ❖ Uluslararası Düzenlemelere Uygunluk: Cihaz, uluslararası standartlara ve düzenlemelere uygun olarak tasarlanmış olup, askeri sistemlerde kullanılabilir.
- ❖ Arızalar Arası Ortalama Süre (MTBF): 750.000 saate kadar güvenilir çalışma ömrü sunar.
- ❖ Kısa Devre Koruması ve ESD Koruması: Elektriksel güvenlik için tasarlanmış entegre koruma.
- ❖ EMI Uyumlu Tasarım: Elektromanyetik girişime karşı koruma sağlar.
- ❖ ASA, ABS ve Karbon Malzeme Kullanımı: Cihaz, dayanıklılığı artırmak ve uzun ömürlü kullanım sağlamak için kaliteli malzemelerle üretilmiştir.

✓ Uyumluluk

- ❖ Tüm büyük işletim sistemleri için destek
Windows®, Linux®, macOS, Android™

3.0 RFIDAX Serisi Mimarisi

3.1 Cihaz Okuyucu Yazıcı Blok Diyagramı

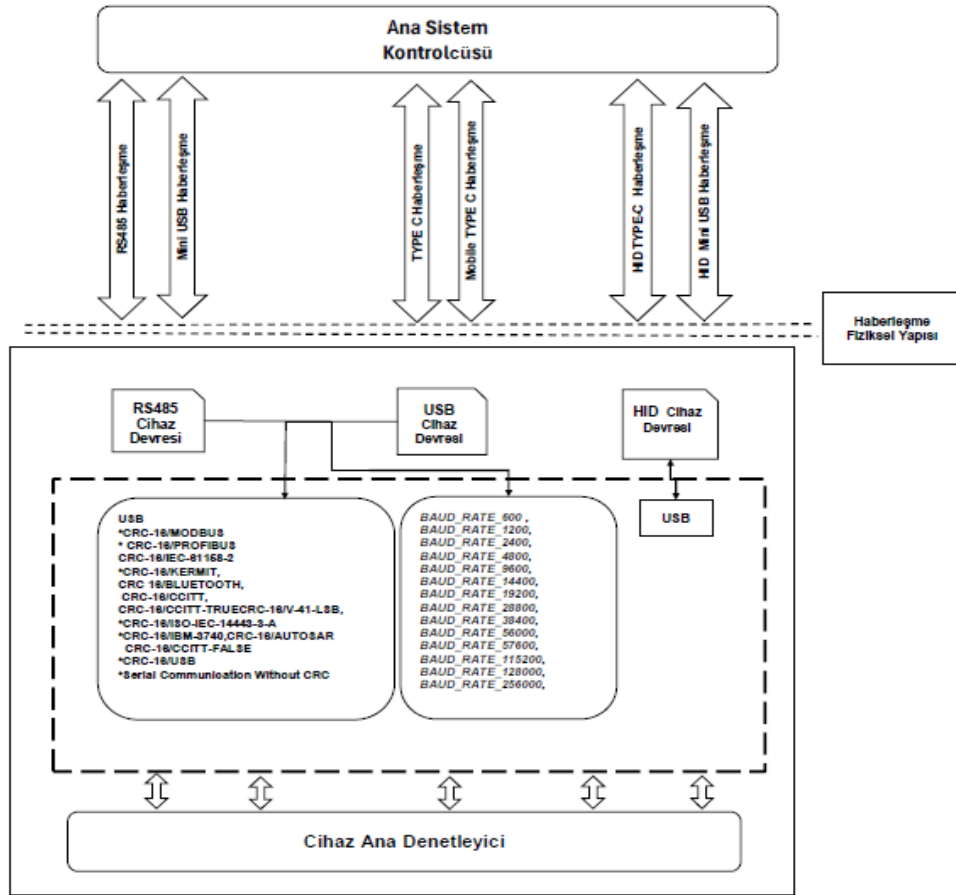


Şekil 1 RFIDAX Okuyucu Blok Diyagramı

3.2 İletişim Protokolü Yapısının Blok Diyagramı

RFIDAX cihazları, RS485, Mini USB ve Type-C gibi farklı fiziksel haberleşme katmanlarını destekleyerek geniş bir bağlantı altyapısı sunar. Cihaz, özellikle endüstriyel ortamlarda ve savunma sanayisinde kullanım için tasarlanmış gelişmiş haberleşme protokolleriyle donatılmıştır.

Bu protokoller, kullanıcı ihtiyaçlarına göre istenilen hızlarda ayarlanabilir ve cihazın çalışması sırasında (runtime) hız ve protokol değişikliği yapılmasına olanak tanır. Bu benzersiz özellik, rakip ürünlere kıyasla önemli bir avantaj sağlar ve sistemi hem esnek hem de yüksek performanslı bir çözüm haline getirir.



Şekil 2 RFIDAX Cihazları Haberleşme Blok Diyagramı



Advanced RFID Card Reader Writer

4.0 RFIDAX Cihazlarının Model ve Cihaz numaraları

Aşağıdaki tabloda RFIDAX cihazlarına ait model ve model kısaltmaları verilmiştir. Harici olarak açıklayıcı bilgi de yer almaktadır. İstedığınız özelliklere ve modlara göre bu tablodan cihaz seçebilirsiniz.

Model Bilgisi	Model numarası	Cihaz açıklaması
Mini USB Serisi	MUSB-RD	Mini USB konnektörü kullanan okuma işlemi yapan cihazdır.
	MUSB-RW	Mini USB konnektörü kullanan okuma ve yazma işlemi yapan cihazdır. Farklı yazılımsal protokolleri destekler.
	HID- MUSB	Mini USB konnektörü kullanan HID cihazdır. Klavye gibi davranır.
Type-C Serisi	TC-RW	Type -C konnektörü kullanan okuma ve yazma işlemi yapan cihazdır. Farklı yazılımsal protokolleri destekler.
	TC-RD	Type -C konnektörü kullanan okuma işlemi yapan cihazdır.
	TC-Mobile	Type -C konnektörü kullanan okuma ve yazma işlemi yapan cihazdır. Farklı yazılımsal protokolleri destekler.
	HID- TYPEC	TYPE-C konnektörü kullanan HID cihazdır. Klavye gibi davranır.
Rs485 Serisi	RS485-RD	Rs485 konnektörü kullanan okuma işlemi yapan cihazdır.
	RS485-RW	Rs485 konnektörü kullanan okuma ve yazma işlemi yapan cihazdır. Farklı yazılımsal protokolleri destekler.

Tablo 1 Cihaz Model numaraları ve model bilgisi



Advanced RFID Card Reader Writer

5.0 Donanım Tasarımı

5.1 USB Bağlantı

RFIDAX markalı cihazların HID USB ve standart USB modelleri, USB standardına uygun olarak USB aracılığıyla bilgisayara bağlanır.

5.1.1. İletişim Parametreleri

RFIDAX, USB 2.0 Standardı'na uygun olarak bilgisayara USB bağlantısı aracılığıyla bağlanır. RFIDAX, tam hız modunda (full-speed mode) çalışır ve bu hız 12 Mbps'dir.

Pin	Sinyal	Görev
1	VBUS	Cihaz için +5 V güç kaynağı sağlar.
2	D-	RFIDAX ilgili cihazı ile bilgisayar arasında veri iletimi için kullanılan diferansiyel sinyal hattı
3	D+	RFIDAX ilgili cihazı ile bilgisayar arasında veri iletimi için kullanılan diferansiyel sinyal hattı
4	GND	Güç kaynağı için referans voltaj seviyesi sağlar

Tablo 2 USB Arayüzü Kablolama

Not: Cihazın USB arayüzü üzerinden düzgün çalışabilmesi için, **cihaz sürücüsünün yüklenmesi gereklidir.**

5.2 RS485 Bağlantı

RFIDAX'ın RS485-RW ve RS485-RD serisi cihazları RS485 fiziksel katmanı üzerinden bilgisayar veya cihaz kontrol cihazları ile iletişim kurabilir. RS485 cihazları iletişim kurabilmesi için bir USB-485 çeviriciye ihtiyaç duymaktadır. Haberleşme half duplex haberleşmedir.

5.2.1. İletişim Parametreleri

RS485 fiziksel katmanı ile iletişim hızı değişiklik gösterebilir ve ayarlanabilir. Ayarlanabilir iletişim hızları ise 600bps,1200bps,2400bps,4800bps,9600 bps,14.4kbps,19.2 kbps,28.8 kbps,38.4 kbps,56 kbps,57.6 kbps,115.2 kbps,128 kbps ,256kbps'dır.

Bu rakip ürünlere göre ayırtıcı bir özelliktir.

Pin	Sinyal	Görev
1	VBUS	Cihaz için +24 V güç kaynağı sağlar.(Talep edilirse +12v ayarlanan modeller vardır.)
2	RS485 -A	RFIDAX ilgili cihazı ile bilgisayar arasında veri iletimi için kullanılan diferansiyel sinyal hattı
3	RS485 -B	RFIDAX ilgili cihazı ile bilgisayar arasında veri iletimi için kullanılan diferansiyel sinyal hattı
4	GND	Güç kaynağı için referans voltaj seviyesi sağlar

Tablo 3 RS485 Arayüzü Kablolama

Not: Cihaz karşınıza aldığınızda en üsteki pin (1 numaralı pin) VCC pindir. En alttaki pin yani led yakın pin ise GND pindir.



Advanced RFID Card Reader Writer

5.3 Type-C Bağlantı

RFIDAX markalı cihazların HID TYPE-C, standart TYPE-C ve mobile modelleri Type-C standardına uygun olarak USB aracılığıyla bilgisayara bağlanır. (TC-RW, TC-RD, TC-Mobile)
Fiziksel katmanı TYPE-C'dir.

5.3.1. İletişim Parametreleri

RFIDAX, USB 3.0 ve USB 2.0 Standardı'na uygun olarak bilgisayara ve mobile cihazlara USB bağlantısı aracılığıyla bağlanır. RFIDAX, tam hız modunda (full-speed mode) çalışır ve bu hız 12 Mbps'dir.

Pin	Sinyal	Görev
A4	VBUS	RFIDAX cihazını kontrol eden cihazın güç hattına göre değişiklik gösterebilir. RFIDAX bunun için uygundur, kendini optimize edebilir. (min. +5V, max.24V) RFIDAX kontrol eden cihaza bağlıdır. Normal PC standartları uygundur.
B9	VBUS	RFIDAX cihazını kontrol eden cihazın güç hattına göre de değişiklik gösterebilir. RFIDAX bunun için uygundur, kendini optimize edebilir. (min. +5V, max.24V) RFIDAX kontrol eden cihaza bağlıdır. Normal PC standartları uygundur.
A9	VBUS	RFIDAX cihazını kontrol eden cihazın güç hattına göre değişiklik gösterebilir. RFIDAX bunun için uygundur, kendini optimize edebilir. (min. +5V, max.24V) RFIDAX kontrol eden cihaza bağlıdır. Normal PC standartları uygundur.
B4	VBUS	RFIDAX cihazını kontrol eden cihazın güç hattına göre değişiklik gösterebilir. RFIDAX bunun için uygundur, kendini optimize edebilir. (min. +5V, max.24V) RFIDAX kontrol eden cihaza bağlıdır. Normal PC standartları uygundur.
A6	DP1	RFIDAX ilgili cihazı ile bilgisayar arasında veri iletimi için kullanılan diferansiyel sinyal hattı
B6	DP2	RFIDAX ilgili cihazı ile bilgisayar arasında veri iletimi için kullanılan diferansiyel sinyal hattı
A7	DN1	RFIDAX ilgili cihazı ile bilgisayar arasında veri iletimi için kullanılan diferansiyel sinyal hattı
B7	DN2	RFIDAX ilgili cihazı ile bilgisayar arasında veri iletimi için kullanılan diferansiyel sinyal hattı
A5	CC1	Sadece RFIDAX'in mobile cihazlarında kullanılır.
B5	CC2	Sadece RFIDAX'in mobile cihazlarında kullanılır.
A1	GND	Güç kaynağı için referans voltaj seviyesi sağlar
B1	GND	Güç kaynağı için referans voltaj seviyesi sağlar
A12	GND	Güç kaynağı için referans voltaj seviyesi sağlar
B12	GND	Güç kaynağı için referans voltaj seviyesi sağlar
SHIELD	SHIELD	Cihaz için SHIELD görevi görür.

Tablo 4 TYPE-C Arayüzü Kablolama

Not: Cihazın TYPE-C arayüzü üzerinden düzgün çalışabilmesi için, **cihaz sürücüsünün yüklenmesi gereklidir.**



Advanced RFID Card Reader Writer

5.4 Temassız Akıllı Kart Arayüzü

RFIDAX cihazları ile temassız kart arasındaki arayüz, ISO 14443 spesifikasyonlarına uygun olarak çalışır. Bununla birlikte, RFIDAX'in pratik işlevselliğini artırmak amacıyla data manipölasyonu, belirli kısıtlamalar veya iyileştirmeler içerebilir.

5.4.1. Taşıyıcı Frekansı

RFIDAX'in taşıyıcı frekansı 13.56 MHz'dir.

5.4.2. Kart Tarama (Polling)

RFIDAX, alan içerisindeki temassız kartları otomatik olarak tarar. ISO 14443 Type A (Parts 1-4), MIFARE® (T=CL), MIFARE Classic, MIFARE Ultralight, MIFARE DESFire EV1, MIFARE Plus, MIFARE Mini ve NFC Tags kartlarını destekler.

5.5 Kullanıcı Arayüzü

5.5.1. Buzzer ve LED

Tek tonlu programlanabilir buzzer ve programlanabilir LED'ler, **temassız arayüzlerin durumunu göstermek** için kullanılır. **Mavi ve yeşil LED, ve buzzer cihaz ve haberleşme durumunu göstermek** için kullanılır.

Yeşil led cihaz hardware olarak sadece revizyon 1 mevcuttur. Mavi led ise revizyon 1 dışındaki cihazlarda kullanılmaktadır. Kırmızı led her versiyonda mevcuttur.

Cihaz Durumu	Buzzer	Kırmızı ve Mavi LED
1. Okuyucu takıldığında	Kapalı	●
2. Bekleme modu (Temassız tarama açık, kart yok)	Kapalı	●
3. Bekleme modu (Tarama kapalı)	Kapalı	●
4. Temassız kart okutulduğunda	1kez Beep	●
5. Temassız kart okutma alanında kaldığında	Kapalı	●
6. Temassız kart kaldırıldığında	Kapalı	●
7. Temassız kart ile iletişim kurulduğunda	kapalı	Hızlı yanıp Sönme >> ● >> ● >> ● >> ●

Tablo 5 Buzzer ve LED Göstergeleri



6.0 Cihaz Yazılımsal Komutları ve Konfigasyonu

Bu bölümde cihazın yazılımsal komut yapısı, gerekli görüldüğü kadar çalışma algoritması, gelen ve giden veri yapısını anlatılacaktır.

Cihazın düzgün çalışması için gönderilecek veri yapısı,

Cihazdan gelen yanıtlar ve bu yanıtların anlamı,

Sistem monitoring yapısı, hata yapısı,

Cihazın desteklediği RFID kartlarla ilgili bütün işlemlerin yapılması,

Cihaz modları anlatılacaktır.

Cihaz komutları ve konfigasyonu ilk başta karmaşık gibi görünebilir ama belirli bir algoritmaya göre ilerlemektedir.

Oldukça basit ve geliştirmeye uygundur.

Cihazın komut yapısı anlatılırken ücretsiz terminaller kullanılacaktır.

Ücretsiz terminallerin kullanılma amacımız RFIDAX cihazlarının kolay bir şekilde yazılımlarınıza entegre etmenize yardımcı olmaktır.

Bütün cihaz haberleşmesinde Heksadesimal değerler kullanılacaktır.

Cihaz komut yapısını anlamak için ilk başta tanımlar ve sistem ile ilgili genel bilgi verilecektir.

6.1 Cihazın Modları ve Açıklamaları

Cihaz modları ve açıklamaları aşağıda verilmiştir.

Bu komut yapısının fonksiyon türleri ve görevleri operasyon modlara göre değişiklik gösterebilir,

Bu değişiklik fonksiyonun aktifliği ve pasifliği seklindedir.

Operasyon modları aşağıdadır;

Cihaz Modları	Cihaz Mod Açıklamaları
MODE_BASIC	Kurumsal kullanım için optimize edilmiştir. Bu mod sadece RFID kart okuma ve bazı basit işlemleri gerçekleştirir. Ticari uygulamalarda kullanılabilir.
MODE_ADVANCED	Bütün okuma ve yazma işlemleri gerçekleştirir. Bütün özellikler açıktır. Ticari, savunma ve uzay sanayinde kullanılır.
MODE_CUSTOM	Müşteriye özel bir moddur. Özel kurumsal müşteriler için gerçekleştirilmiştir.
MODE_ENTERPRISE	Kurumsal kullanım için optimize edilmiştir ve ticari uygulamalar için gelişmiş özellikler sunar.

Tablo 6 Cihaz Modları ve açıklamaları

CUSTOM mode tamamen özel müşteri isteğine göre tasarlanmış bir moddur. RFIDAX firmasıyla iletişime geçen müşteriler için hazırlanmış bir moddur. Özellikler müşteriye göre değiştirilmiştir, tamamen özel bir moddur.

ENTERPRISE mode ise var olan özellikleri kullanarak **mevcut yapıyı değiştirmeden kurumsal müşteriye** sunulan moddur. ADVANCED modun özellik açısından daraltılmış halidir. Özellikler değiştirilmemiştir.

Bu dokümanda CUSTOM mode açıklanmayacaktır.

Dokümanlardaki özellikler hangi modda aktif olarak kullanıldığı yazmıştır.

6.2 Veri Bütünlüğü Modları ve Açıklamaları

CRC'nin aktif olup olmadığını belirleyen moddur. Bu modlar aşağıdaki tabloda verilmiştir.

Veri Bütünlüğü Modları	Veri Bütünlüğü Modları Açıklamaları
DATA_INTEGRITY_CHECK	Kurumsal kullanım için optimize edilmiştir. CRC'nin aktif olduğunu gösterir. (MODE_ADVANCED, MODE_ENTERPRISE)
NO_DATA_INTEGRITY_CHECK	Kurumsal kullanım için optimize edilmiştir. CRC'nin aktif olmadığını gösterir. (MODE_BASIC)

Tablo 7 Veri Bütünlüğü Modları ve Veri Bütünlüğü Modları Açıklamaları

6.2.1 Veri bütünlüğü Pasif Olduğu Durum

Bu modda CRC aktif değildir. Mode Basic bu modu aktif olarak kullanır.

6.2.2 Veri bütünlüğü Aktif Olduğu Durum

DATA INTEGRITY CHECK etkin olduğunda, çeşitli CRC modları aktif hale gelir.

MODE_ADVANCED ve MODE_ENTERPRISE aktif olarak kullanır.

Aşağıda listelenen bu CRC modları, **gelişmiş ve profesyonel çözümler sunar**. Kullanıcı, ihtiyacına uygun herhangi bir CRC modunu tercih edebilir.

Varsayılan olarak, CRC aktif olduğunda **PROTOCOL_CRC16_CCITT** yapısı kullanılır.

Bütün CRC modları CRC-16'dır.

Crc Modları	Modun desteklediği alt protokoller
PROTOCOL_CRC16_USB	CRC-16/USB
PROTOCOL_CRC16_PROFIBUS	CRC-16/PROFIBUS CRC-16/IEC-61158-2
PROTOCOL_CRC16_MODBUS	CRC-16/MODBUS MODBUS
PROTOCOL_CRC16_KERMIT,	CRC-16/KERMIT CRC-16/BLEETOOTH CRC-16/CCITT CRC-16/CCITT-TRUE CRC-16/V-41-LSB CRC-CCITT KERMIT
PROTOCOL_CRC16_CCITT	CRC-16/IBM-3740 CRC-16/AUTOSAR CRC-16/CCITT-FALSE
PROTOCOL_CRC16_ISO_IEC14443_A,	CRC-16/ISO-IEC-14443-3-A CRC-A

Tablo 8 CRC modları ve modların desteklediği alt CRC modları

6.3 Cihazların Haberleşme Türleri

Cihazların haberleşme türleri aşağıdadır. Bu türler hardware yapılarına göre sıralanmıştır. Bu yapıya göre RFIDAX cihazları isimlendirilmiştir. Model ve model numaraları buna göre isimlendirilmiştir. Bütün cihaz modları bu haberleşme türleri destekler.

Haberleşme Türü	Haberleşme Türü Açıklaması
COMM_USB	USB donanımı kullanan cihazdır. Satışta, aktiftir.
COMM_RS485	RS485 donanımı kullanan cihazdır. Satışta, aktiftir.
COMM_TYPE_C	TYPE C donanımı kullanan cihazdır. Satışta, aktiftir.
COMM_MOBILE	TYPE C donanımı kullanan cihazdır. Satışta, aktiftir.
COMM_ETHERNET	Geliştirilmektedir, satışta değildir.
COMM_WIFI	Geliştirilmektedir, satışta değildir.

Tablo 9 Haberleşme Türleri ve Haberleşme Türleri Açıklamaları

6.4 Cihaz Durum Mesajı, Hata Kodları ve Açıklamaları

Bu bölümde, RFIDAX cihazında meydana gelebilecek hata kodları, açıklamaları ve olası çözümleri listelenmiştir.

Önemli Bilgiler:

Tüm hata kodları 4 bayt aralıklarla tanımlanmıştır. Bu, ileride yeni kodların eklenmesini kolaylaştırır.

Hata kodları HEX formatında gösterilmektedir. Cihaz haberleşmesi sırasında bu formatta dönecektir.

Her hata kodunun altında "Olası Çözüm" bölümü eklenmiştir. Kullanıcı sorunu gidermek için izlenmesi gereken adımları görebilir.

6.4.1 Genel Sistem Durumu ve Hataları

Bu hata kodları, cihazın genel işleyişi ve haberleşmesi sırasında karşılaşılabilecek temel hataları temsil eder.

Hata Kodu	Hata Açıklaması	Olası Çözüm
0x0000	SUCCESS – İşlem başarılı.	Hata yok, işlem başarıyla tamamlandı.
0x0001	ERR_UNKNOWN_COMMAND – Bilinmeyen komut.	Geçersiz bir komut gönderilmiş olabilir, desteklenen komutları kontrol edin.
0x0002	ERR_CRC – CRC hatası.	Veri paketi hatalı gelmiş olabilir, uygun CRC modunda olmayabilir veya master cihaz CRC yanlış hesaplayıp yollamış olabilir.
0x0003	ERR_ADDRESS_MISMATCH – Cihaz adresi uyumsuzluğu.	Doğru cihaza komut gönderdiğinizizi kontrol edin.
0x0004	ERR_BUFFER_OVERFLOW – Bellek taşması.	Komutun veri uzunluğunu kontrol edin ve uygun şekilde yeniden gönderin.

Tablo 10 Genel Sistem Hataları ve Açıklamaları

6.4.1.1 Genel Sistem Hataları Ek Açıklama

ERR_ADDRESS_MISMATCH bu hata cihaz adresinin yanlış olduğunda alabileceğiniz bir hatadır. Cihaz adresini kontrol ediniz.

ERR_BUFFER_OVERFLOW hatası geldiğinde RFIDAX cihazına gönderilen buffer boyutuna bakınız. Bu buffer boyutu 1024 byte fazla olmamalıdır.

Eğer buffer boyutu 1024 byte büyük ise **ERR_BUFFER_OVERFLOW** hatası alınır.

6.4.2 Zaman Aşımı ve Veri Hataları

Hata Kodu	Hata Açıklaması	Olası Çözüm
0x0008	ERR_TIMEOUT – Zaman aşımı.	Cihazdan yanıt alınamıyor, bağlantıyı kontrol edin.
0x000C	ERR_INVALID_DATA – Geçersiz veri.	Gönderilen verinin doğruluğunu kontrol edin.

Tablo 11 Zaman Aşımı ve Veri Hataları ve Açıklamaları

6.4.2.1 Zaman Aşımı ve Veri Hataları Ek Açıklama

ERR_TIMEOUT bu hata nadiren görünür, lütfen cihaza gönderilen komutun süresini dikkat ediniz.

ERR_INVALID_DATA – bu hata nadiren görünür, lütfen cihaza gönderilen komutun yapısına dikkat ediniz.

6.4.3 LED ve Göstergeler ile İlgili Hatalar

Bu hata kümesi ledler ile ilgili durumu belirtir.

Hata Kodu	Hata Açıklaması	Olası Çözüm
0x0010	ERR_INVALID_LED_STATE – Geçersiz LED durumu.	LED'in açık veya kapalı olup olmadığını doğru kontrol edin.
0x0014	ERR_LED_NUMBER_INVALID – Geçersiz LED numarası.	Kullanılan LED ID'sini doğru olup olmadığını kontrol edin.
0x0018	ERR_INVALID_TOGGLE – Geçersiz toggle işlemi.	LED'in geçerli bir duruma sahip olup olmadığını kontrol edin.

Tablo 12 LED ve Göstergeler ile İlgili Hatalar ve Açıklamaları

6.4.3.1 LED ve Göstergeler ile İlgili Hataların Ek Açıklaması

ERR_INVALID_LED_STATE bu hata hatalı led durumu ile ilgilidir. RFIDAX cihazında led iki durumda kontrol edilir.

Led durumu hex olarak 0x00 veya 0x01 olarak belirtilmiştir. Harici durumda bu hata geri döner.

ERR_LED_NUMBER_INVALID bu hata hatalı led numarası ile ilgilidir. RFIDAX cihazında Led numarası hex olarak 0x00 veya 0x03 olarak belirtilmiştir. Harici durumda bu hata geri döner.

Not: Led numarası 0x02 değeri Custom mode için kullanılır. BASIC ve ADVANCED mode kullanılmaz.

6.4.4 RFID Kart Okuma/Yazma Hataları

Bu hata kümesi, RFID kart okuma ve yazma işlemleri sırasında karşılaşılabilecek hataları içermektedir. Kartın doğru şekilde okutulması ve verilerin düzgün bir biçimde işlenmesi için bu hata kodları dikkatlice incelenmelidir. Kart algılanmadığında, UID okunamadığında veya yetkilendirme başarısız olduğunda, buradaki hata kodlarına bakarak sorunun kaynağını belirleyebilirsiniz.

Bu hatalar genellikle kartın fiziksel konumu, yetkilendirme anahtarı veya blok erişim izinleri ile ilgilidir.

Bazı hatalar yazılım ayarlarından kaynaklanabilir, bu nedenle sistemin doğru konfigüre edildiğinden emin olun.

Flash belleğe yazma ile ilgili hatalar, sistemin veri kaydetme işlemlerini etkileyebilir, bu nedenle gerekli önlemler alınmalıdır.

Eğer RFID kart okuma veya yazma işlemlerinde hata alıyorsanız, aşağıdaki hata kodlarını ve olası çözümleri takip ederek sorunu giderebilirsiniz.

Hata Kodu	Hata Açıklaması	Olası Çözüm
0x001C	ERR_CARD_UID_READ_FAILED – Kart UID okunamadı.	Kartı yeniden okutun veya bağlantıyı kontrol edin.
0x0020	ERR_CARD_NOT_FOUND – Kart bulunamadı.	Kartın okuyucu alanına uygun şekilde yerleştirildiğini kontrol edin.
0x0024	ERR_ATQA_VALUE_NOT_READ – ATQA değeri okunamadı.	Kartın türünü kontrol edin, desteklenen kart olup olmadığını doğrulayın.
0x0028	ERR_CARD_NOT_SELECTED – RFID kart seçilemedi.	Kartı farklı bir pozisyonda deneyin veya tekrar okutun.
0x002C	ERR_INVALID_KEY_TYPE – Geçersiz anahtar tipi.	Doğru anahtarın kullanıldığını kontrol edin.
0x002D	ERR_AUTHENTICATION_FAILED – Blok yetkilendirme başarısız.	Anahtar veya kimlik doğrulama işlemlerinin doğru olup olmadığını kontrol edin.
0x0030	ERR_UNKNOWN_RFID_SUBCOMMAND – Bilinmeyen alt RFID komutu.	Desteklenen alt komutları kontrol edin.
0x0034	ERR_INVALID_BLOCK_RANGE – Hatalı blok aralığı.	Blok numarasının doğru olup olmadığını kontrol edin.
0x0038	ERR_INVALID_AUTH – Kimlik doğrulama başarısız.	Kullanıcı yetkisini ve anahtar doğruluğunu kontrol edin.
0x004C	ERR_BLOCK_READ_FAILED – Blok okuma hatası.	Kartın okuyucu ile doğru şekilde temas ettiğinden emin olun.
0x0050	ERR_DATA_LENGTH_EXCEEDED – Veri uzunluğu sınırını aştı.	Gönderilen veri miktarını azaltın.
0x0054	ERR_BLOCK_WRITE_FAILED – Blok yazma hatası.	Kartın yazılabilir olup olmadığını kontrol edin.
0x0058	ERR_FLASH_KEY_OPERATION – Geçersiz Flash anahtar işlemi.	Flash belleğe yazma işlemini doğrulayın.
0x0060	ERR_FLASH_ID_RECORD_FAILED – ID kayıt işlemi başarısız.	Flash bellekte yeterli alan olup olmadığını kontrol edin.
0x001D	ERR_UNKNOWN_CARD_TYPE-Bilinmeyen kart	Bilinmeyen kart tipi algılandı
0x002E	ERR_INVALID_AUTH_TYPE- Geçersiz AUTH tipi	Kimlik doğrulama türü geçersiz
0x002F	ERR_FORMAT_FAILED- Kart format hatası	Kart Formatlama sırasında hata oluştu
0x0039	ERR_ALL_SECTOR_UPDATE_FAILED- trailer güncellenememesi	Hiçbir trailer güncellenmediyse hata döner

Tablo 13 RFID Kart Okuma/Yazma Hataları ve Açıklamaları



Advanced RFID Card Reader Writer

6.4.4.1 RFID Kart Okuma/Yazma Hatalarının Ek Açıklaması

ERR_CARD_UID_READ_FAILED: kartın bulunduğunu fakat ID okunamadığını belirtir. Kartın sağlamlığını veya okuma alanına doğru yerleştirildiğini kontrol ediniz.

ERR_UNKNOWN_RFID_SUBCOMMAND: cihazın komut yapısı belirli bir düzene göre tasarlanmıştır. Cihaza gönderilen komutun hatalı olduğunu belirtir.

ERR_DATA_LENGTH_EXCEEDED: rfid karta yazma işleminde veri uzunluğu istenmektedir. Bu veri uzunluğunun karta yazılmak istenen data ile uyumsuz olduğu belirtir.

ERR_FLASH_KEY_OPERATION: cihaz RFID kartı okurken ve yazarken belirli bir **KeyA ve KeyB**'i kullanır. Bu keyA ve KeyB'i RFIDAX cihazları akılda tutabilir, cihaz içerisine kaydedebilir. Bu kaydetme işlemleri sırasında (cihaz içine okuma/yazma) bir hata oluşup oluşmadığını kontrol eder. Bir hata durumunda bunu belirtir.

ERR_FLASH_ID_RECORD_FAILED, RFIDAX cihazları RFID kartların ID'lerini manuel olarak hafıza tutabilir. Bu ID'ler ile okuma yazma ve güncelleme işlemlerini destekler. Bu işlemler sırasında bir hata olursa bu hatayı iletir.

ERR_UNKNOWN_CARD_TYPE Cihaza sunulan kart tipi desteklenmiyor veya algılanamıyor. Farklı bir kart tipi deneyiniz.

ERR_INVALID_AUTH_TYPE Cihaza gönderilen kimlik doğrulama türü (AuthA / AuthB) geçersiz veya sistemle uyumsuz. Komutu doğrulayınız.

ERR_FORMAT_FAILED Kart formatlama işlemi sırasında hata oluşmuştur. Kart tipi, anahtarlar veya erişim bitleri uygun değilse bu hata oluşabilir.

ERR_ALL_SECTOR_UPDATE_FAILED Kart üzerindeki hiçbir sector trailer başarıyla güncellenemediyse bu hata döner. Muhtemel nedenler arasında kartın yazma korumalı olması veya kimlik doğrulamanın başarısız olması yer alır.

6.4.5 İleri Seviye Hatalar ve Sistem Yönetimi

Bu hata kodları, gelişmiş modlar, bellek yönetimi, haberleşme, protokol doğrulama ve sistem güvenliği ile ilgilidir. Kullanıcı, bu hata kodlarını kontrol ederek cihazın doğru çalışmasını sağlayabilir ve sorunları gidermek için gerekli önlemleri alabilir.

Bu hata kodları, genellikle gelişmiş cihaz ayarları ve sistem yönetimiyle ilgilidir.

Hata mesajları, cihazın donanım veya yazılım seviyesinde bir problem yaşadığını gösterebilir.

Hata durumunda, ilgili çözüm önerilerini uygulayarak sorunu giderin.

Hata Kodu	Hata Açıklaması	Olası Çözüm
0x0064	ERR_ADVANCED_MODE_NOT_SUPPORTED – Komut bu modda desteklenmiyor.	Kullanılan komutun uygun modda olup olmadığını kontrol edin
0x0068	ERR_CONTACT_MANUFACTURER – Üretici ile iletişime geçin.	Cihaz arızalı olabilir veya yazılım güncellemesi gerekebilir. Yanlış modda cihaz alınmış olabilir. RFIDAX teknik destek ekibine başvurun.
0x0070	ERR_INVALID_DEVICE_INFO_COMMAND – Geçersiz cihaz bilgi komutu.	Cihaza bilgi sorgulama komutu (device info request) gönderirken doğru parametreleri kullanın.
0x0071	ERR_RESET_CARD_NOT_READ – Reset kartı algılanamadı. Lütfen reset kartını tanıttın.	Cihaza doğru reset kartının okutulduğundan emin olun.
0x0072	ERR_INVALID_DEVICE_ADDRESS – Geçersiz cihaz adresi.	Kullanılan cihaz adresini değiştirmek için kullanılır. İlgili sınırı belirler.
0x0073	ERR_FLASH_WRITE_FAILURE – Flash bellek yazma hatası.	Flash bellek hasarlı olabilir veya dolu olabilir. Flash bellek temizleme işlemi yapın ve tekrar deneyin.
0x0074	ERR_INVALID_PROTOCOL – Geçersiz protokol hatası.	Kullanılan CRC haberleşme protokolünün doğruluğunu kontrol edin.
0x0075	ERR_INVALID_BAUD_RATE – Geçersiz baud rate hatası.	Kullanılan baud rate (haberleşme hızı) cihazın desteklediği hızlarla uyumlu olmalıdır.
0x0076	ERR_INVALID_FORMAT_FLAG – Hatalı format flag değeri.	Gönderilen komut içindeki flag (bayrak) değerinin geçerli olup olmadığını kontrol edin. Yanlış formatta bir veri göndermeyin.
0x0077	ERR_INVALID_HEX_DATA – Geçersiz HEX formatı.	HEX formatında gönderilen verinin geçerli olduğundan emin olun. Sadece 0-9 ve A-F karakterlerinden oluşan HEX değerler kullanın.
0x0078	ERR_FLASH_ERASE_FAILURE – Flash bellek silme hatası.	Flash bellekteki veri silinememiş olabilir. Bu hata alındığında tekrar silme işlemi yapın.
0x0079	ERR_UNKNOWN_RESET_FACTORY_CMD – Bilinmeyen reset factory komutu.	Fabrika ayarlarına geri dönüş komutundaki hatalı alt komut gönderildiğinde bu hata alınır. Cihazın desteklediği reset komutlarını kullanın ve doğru parametreleri gönderdiğinizden emin olun.

Tablo 14 İleri Seviye Hatalar ve Sistem Yönetimi ve Açıklamaları



Advanced RFID Card Reader Writer

6.4.5.1 İleri Seviye Hatalar ve Sistem Yönetimi Hatalının Ek Açıklamaları

ERR_ADVANCED_MODE_NOT_SUPPORTED bu hata alındığında kullanılan komutun uygun modda olup olmadığını kontrol edin. Eğer **MODE_BASIC** veya **MODE_ENTERPRISE** içindeyseniz, komutun sadece **MODE_ADVANCED** için geçerli olabileceğini unutmayın.

ERR_RESET_CARD_NOT_READ, RFIDAX cihazları çok gelişmiş bir haberleşme protokolüne sahiptir. Bu gelişmişlik ve çok varyantlı sistemlerde Reset tag ihtiyaç duyulmaktadır. Bu tag görevi ise cihazı fabrika ayarlarına çevirmektir. Bu tag okunması sırasında karşılaşılan bir hatadır. Cihaza doğru reset kartının okutulduğundan emin olun. Kartın okuma alanı içinde olduğundan ve manyetik alanın yeterli olduğundan emin olun.

ERR_INVALID_DEVICE_ADDRESS bu komut cihazın adresini değiştirmek istendiğinde bir hata ile karşılaşıldığında gelir. Yeni bir cihaz adresinde 0 ile 255 arasında olmalıdır.

ERR_FLASH_WRITE_FAILURE bu hata Flash bellek hasarlı olabilir veya dolu olabilir. **Flash bellek temizleme işlemi yapın ve tekrar deneyin.** Eğer sorun devam ederse üreticiyle iletişime geçin.

ERR_INVALID_PROTOCOL bu hata yanlış bir CRC protokolün seçildiği gösterir. CRC protokol seçimi hex olarak 0x00'den 0x05 kadardır, protokol seçiminde bunun haricinde bir değer cihaza gönderilirse bu hata alınır.

ERR_INVALID_BAUD_RATE bu hata yanlış **baud rate** seçim parametresinden kaynaklanır. **Baud rate** seçimi hex olarak 0x00'den 0x0D kadardır **Baud rate** seçiminde bunun haricinde bir değer cihaza gönderilirse bu hata alınır.

ERR_INVALID_FORMAT_FLAG, **Format** flag değeri hex olarak 0x00 veya 0x01 olmalıdır. Bunun haricinde bir değer alırsa bu hata döner.

6.5 Cihaz Sistem İzleme Döngüsü (System Monitor Loop)

Bu bölümde, RFIDAX cihazlarının sistem izleme (monitoring) mekanizmasının nasıl çalıştığı açıklanmaktadır. Cihaz, kartları sürekli olarak tarayarak belirli işlemleri gerçekleştirir. Kart tanımlama, fabrika ayarlarına sıfırlama, buzzer ve LED bildirimleri gibi süreçler bu sistem izleme döngüsü içinde gerçekleşir.

6.5.1 Sistem İzleme Döngüsünün Çalışma Prensipleri

RFIDAX cihazları, sürekli olarak kart algılama ve sistem durumunu izleme işlemi yapar. Cihaz, kart okutulduğunda veya kaldırıldığında LED, buzzer ve bellek işlemlerini otomatik olarak yönetir.

Sistem izleme döngüsü şu işlemleri gerçekleştirir:

- 1-Kart okutulup okutulmadığını sürekli kontrol eder.
- 2-Kartın daha önce okutulup okutulmadığını kontrol eder ve gereksiz tekrar okumaları engeller.
- 3-Kartın ID'sini bellekte saklanan değer yani Format kartıyla aynı ise fabrika ayarlarına sıfırlama yapar.
- 4-Eğer yeni bir kart okunmuşsa veya tekrar okutulma işlemi 500ms aralığından fazla ise buzzer ve LED sinyali ile bildirim yapar. Master cihaza otomatik mesaj atar.
- 5-Kart hala cihaz üzerindeyse, LED yanıp sönmeye devam eder.
- Bu ledler revizyon 1 cihazında kırmızı -yeşil renktedir, Revizyon 2 cihazında de ise kırmızı mavi renktedir.
- 6- Kart kaldırıldığında, LED ve buzzer sıfırlanarak cihaz normal çalışma moduna döner.
- 7-Harici olarak eğer master cihazdan bir komut gelirse de anlık olarak cevap verir, bu cevap verme aşamasında ise herhangi bir veri kaybı olmaz.
- 8-Bu işlemlerin sonunda ise master cihaza sistem monitoring mesajını otomatik iletir.
- Bu cevap ayrıntılı bir şekilde anlatılmıştır. Lütfen bölüm [6.6 Kart okutulduğunda Cihazdan Otomatik Gelen Cevap \(Master cihaza otomatik gönderilen sistem monitoring mesajı\)](#) bakınız.
- Bu mekanizma, cihazın sürekli olarak çalışmasını ve anlık değişikliklere tepki vermesini sağlar.

6.5.2 Kart Algılama ve Karşılaştırma Süreci

Cihaz, sürekli olarak kart taraması yaparak aşağıdaki işlemleri gerçekleştirir:

Kart tespit edildiğinde, önceki okunan kart ID'si ile karşılaştırma yapılır.

Eğer yeni bir kart ise, buzzer çalıştırılır ve LED bildirimi başlatılır.

Kart ID'si flash bellekte saklanan ID ile karşılaştırılır.

Eğer ID'ler eşleşirse, cihaz fabrika ayarlarına sıfırlanır.

Kart Algılama Senaryoları:

Kart daha önce okutulmadıysa veya farklı bir kart okutulduysa, buzzer aktif edilir ve LED yanıp sönmeye başlar. **Master cihaza sistem monitoring mesajını otomatik iletir.**

Kart ID'si bellekte saklanan (format ID eşleşir ise) ID ile eşleşirse, cihaz fabrika ayarlarına sıfırlanır.

Kart kaldırıldığında, LED ve buzzer sıfırlanarak cihaz bekleme moduna geçer.

6.5.2.1 LED ve Buzzer Bildirimleri

LED ve buzzer kullanımı şu şekilde yönetilir:

Durum	Buzzer	LED
Yeni kart okutuldu	1 kez bip sesi	LED yanıp sönmeye başlar
Kart hala cihaz üzerinde	Sessiz	LED yanıp sönmeye devam eder
Kart ID'si bellekteki format ID ile eşleşti	Sessiz	Cihaz fabrika ayarlarına sıfırlanır
Kart kaldırıldı	Sessiz	LED sabit yanar ve sonra söner

Tablo 15 LED ve Buzzer Bildirimleri

Bunlar, kullanıcının cihazın durumu hakkında anlık geri bildirim almasını sağlar.

6.5.3 Kart Kaldırıldığında Yapılan İşlemler

Eğer bir kart okutulduktan sonra kaldırılırsa:

LED yanıp sönmeye durdurulur.

Kartın mevcut olmadığı belirtilmek için LED sabit yanar.

Buzzer sıfırlanır ve sessiz kalır.

Bu işlem, cihazın yeni bir kart beklediğini ve tekrar okutulabilecek duruma geldiğini gösterir.

6.5.4 Flash Bellek ile Çalışma

Cihaz, flash bellekten veri okuyarak belirli işlemleri otomatik olarak yönetebilir.

Flash bellekte saklanan **format ID** ile okutulan kart ID'si eşleşirse, fabrika ayarlarına sıfırlama işlemi başlatılır.

Bellekteki veri ile kart ID'si eşleşmezse, kart yeni olarak değerlendirilir ve standart işlem devam eder.

Bu mekanizma, önceden belirlenmiş özel kartlarla cihazın yönetilmesine olanak tanır (örneğin, belirli bir kart okutulduğunda cihazın sıfırlanması).

6.5.5 Sistem İzleme Döngüsünün Önemi

Cihazın otomatik olarak kartları algılamasını ve işlemlerini sağlar.

Bellekte kayıtlı ID ile karşılaştırarak fabrika ayarlarına sıfırlama işlemi yapılabilir.

Kullanıcının LED ve buzzer ile anlık geri bildirim almasını sağlar.

Cihazın sürekli olarak çalışmasını ve anlık değişikliklere tepki vermesini garanti eder.

Bu sistem, RFID kartların sürekli olarak izlenmesini ve cihazın kullanıcıya gerekli bildirimleri sağlamasını mümkün kılar.

6.6 Kart okutulduğunda Cihazdan Otomatik Gelen Cevap

(Master cihaza otomatik gönderilen sistem monitoring mesajı)

Bu bölümde, **RFID kart okutulduğunda cihaz tarafından otomatik olarak gönderilen cevap mesajı** detaylı bir şekilde açıklanmaktadır. Bu mesaj, **master cihaza** (bilgisayar, kontrol ünitesi vb.) otomatik olarak iletilir ve kart bilgileri ile cihazın mevcut durumu hakkında bilgi içerir.

Veri Boyutu:

CRC Aktif (DATA_INTEGRITY_CHECK) olduğunda mesaj toplam 32 byte'tır.

CRC Pasif (NO_DATA_INTEGRITY_CHECK) olduğunda mesaj toplam 30 byte'tır.

6.6.1 Tanımlar ve Veri Yapısı

Bu mesajın amacı, **kartın başarılı bir şekilde okutulduğunu bildirmek** ve cihazın mevcut yapılandırmasını paylaşmaktır. Aşağıdaki bileşenlerden oluşur:

- **Başlangıç Baytı (0xAA):** Mesajın başlangıcını belirtir. Heksadesimal değerlerden oluşturulur. Her zaman sabittir. RFIDAX cihazlarının 13.56 Mhz de çalışan cihazları için geçerlidir. Ofset numarası 0'dır.
- **Cihaz Adresi:** Master cihazın, belirli bir RFID okuyucu ile iletişim kurmasını sağlar. 1 ile 255 arasında Device adres gelir. Fabrika ayarı olarak cihazların ilk adresi 01'dir. Kullanıcı isterse sonradan değiştirebilir. Ofset numarası 1'dir.
- **Mesaj Türü:** Bu mesajın sistem monitör mesajı olduğunu gösterir. Sistem monitör mesaj türü her zaman 0x01'dir. Sistem monitör mesaj türü sabittir. (Lütfen [6.7.2.1 Message Type Bilgisi](#) için bakınız) Ofset numarası 2'dir
- **Donanım ve Yazılım Sürümleri:** Cihazın güncel donanım ve yazılım versiyonlarını içerir. Bu bilgiler önemlidir. Önemli olma sebebi ise cihaz firmware güncelleme, cihaz yetenekleri vb. kullanıcıya ve üreticiye bilgi verir. Ofset numarası 3-10 arasındadır. İlgili ofset bilgileri aşağıdadır.

Ofset 3-4	Donanım Versiyonu (Major)
Ofset 5-6	Donanım Versiyonu (Minor)
Ofset 7-8	Yazılım Versiyonu (Major)
Ofset 9-10	Yazılım Versiyonu (Minor)

Bu bilgiler cihaz geliştiği sürece değişir.
- **Derleme Numarası ve Tarihi:** Yazılımın sürüm takibini sağlar. Cihaz ile ilgili son güncellemeler için derleme numarası ve tarih bilgisi verir. Ofset numarası 11-20 arasındadır. İlgili ofsetler aşağıdadır.

Ofset 11-14	Derleme Numarası
Ofset 15-20	Derleme Tarihi

Bu bilgiler cihaz geliştiği sürece değişir.

- **Haberleşme Modu:** Cihazın hangi haberleşme modunu kullandığını belirtir (USB, RS485, Type-C vb.). Ofset numarası 21'dir. Bu ofset numarasındaki hex değer cihazın haberleşme modunu verir.

Haberleşme Türleri	Hex Değeri
COMM_USB	0x00
COMM_RS485	0x01
COMM_TYPE_C	0x02
COMM_MOBILE	0x03
COMM_ETHERNET	0x04
COMM_WIFI	0x05

Tablo 16 Haberleşme Türleri

- **CRC Modu:** Veri bütünlüğünü kontrol eden hata doğrulama mekanizmasıdır. Ofset numarası 22'dir. Bu ofset numarasındaki hex değer cihazın veri bütünlüğü (CRC aktif mi pasif mi bilgisi) modunu verir.

VERİ BÜTÜNLÜĞÜ MODLARI	Hex Değeri
NO_DATA_INTEGRITY_CHECK	0x00
DATA_INTEGRITY_CHECK	0x01

Tablo 17 Veri bütünlüğü modları

- **Operasyon Modu:** Cihazın çalışma modunu belirtir (Basic, Advanced, Enterprise vb.). Ofset numarası 23'tür. Bu ofset numarasındaki hex değer cihazın çalışma modunu verir.

OPERASYON MODLARI	Hex Değeri
MODE_BASIC	0x00
MODE_ADVANCED	0x01
MODE_ENTERPRISE	0x02
MODE_CUSTOM	0x03

Tablo 18 Çalışma (operasyon Modları) Modları

- Ofset numarası 24 olan byte yedek alandır. Gelecekteki kullanım için ayrılmıştır.
- **Kart Kimlik Numarası (UID):** Okutulan kartın benzersiz kimlik numarası. Okutulan RFID kartının benzersiz kimlik numarası bilgisi verir. Ofset numarası 25-28 arasındır.
- Ofset numarası 29 olan byte rezerve alandır. Üretici için ayrılmıştır.
- **CRC Kontrol Değeri:** Veri bütünlüğünü doğrulamak için kullanılan hata kontrol baytları. Gelen mesajın ilgili CRC hesaplar, fabrika çıkış CRC hesaplama metottu CRC-16 CCITT budur. Ofset numarası 30-31 arasındır.

6.6.2 Byte Byte Açıklama

Aşağıda, cevap mesajının her byte'ının anlamı detaylı olarak açıklanmıştır:

Byte Numarası	Byte Değeri (Hex)	Byte Açıklaması	Genel açıklama
0 byte	0xAA	Başlangıç Baytı	Her zaman sabittir
1 byte	Değişken değer	Cihaz Adresi	Değişken olabilir
2 byte	0x01	Mesaj Türü	Sistem monitör ekran mesaj türü (Her zaman Sabittir)
3 byte	Değişken değer	Hardware Versiyon (Major)	Hardware major bilgileri için 2 byte alan ayrılıştır (3-4 byte)
4 byte	Değişken değer	Hardware Versiyon (Major)	
5 byte	Değişken değer	Hardware Versiyon (Minor)	Hardware minor bilgileri için 2 byte alan ayrılıştır (5-6 byte)
6 byte	Değişken değer	Hardware Versiyon (Minor)	
7 byte	Değişken değer	Software Versiyon (Major)	Software major bilgileri için 2 byte alan ayrılıştır (7-8 byte)
8 byte	Değişken değer	Software Versiyon (Major)	
9 byte	Değişken değer	Software Versiyon (Minor)	Software minor bilgileri için 2 byte alan ayrılıştır (9-10 byte)
10 byte	Değişken değer	Software Versiyon (Minor)	
11 byte	Değişken değer	Build Numarası	Build numarası bilgileri için 4 byte alan ayrılmıştır (11-14 byte)
12 byte	Değişken değer		
13 byte	Değişken değer		
14 byte	Değişken değer		
15 byte	Değişken değer	Build Tarihi	Build tarihi bilgileri için 6 byte alan ayrılmıştır(15-20byte)
16 byte	Değişken değer	Build Tarihi	
17 byte	Değişken değer	Build Tarihi	
18 byte	Değişken değer	Build Tarihi	
19 byte	Değişken değer	Build Tarihi	
20 byte	Değişken değer	Build Tarihi	
21 byte	Değişken değer	Haberleşme Mode	Haberleşme modu için 1 byte alan ayrılmıştır
22 byte	Değişken değer	CRC Mode	CRC modu için 1 byte alan ayrılmıştır
23 byte	Değişken değer	Operation Mode	Operation Mode için 1 byte alan ayrılmıştır
24 byte	Sabit değer 0x00	Yedek Alan	Yedek Alan ileride geliştirmek için yedek olarak bırakılmıştır.
25 byte	Değişken değer	Card id değeri	Bu 4 byte kartın id değeridir
26 byte	Değişken değer	Card id değeri	
27 byte	Değişken değer	Card id değeri	
28 byte	Değişken değer	Card id değeri	
29 byte	Değişken değer		Rezerve byte lütfen kullanmayın (kart üreticisi ile ilgilidir)
30 byte	Değişken değer	CRC Kontrol Değeri	CRC-16 CCITT algoritması ile hesaplanmış 2 Byte bir kontrol değeri (30-31byte)
31 byte	Değişken değer	CRC Kontrol Değeri	

Tablo 19 Sistem monitoring yapısının Byte Byte Açıklama

6.6.2.1 Byte Byte Açıklama Özeti

Yukarıda verilen Sistem monitör mesajının özeti aşağıdaki tablodadır.

Ofset (Byte)	Hex Değer	Byte Açıklaması	Detaylı Açıklama
0	0xAA	Başlangıç Baytı	Veri paketinin başlangıç baytı, her zaman sabittir.
1	Değişken	Cihaz Adresi	İletişim yapılan cihazın adresini belirten değişken bayt.
2	0x01	Mesaj Türü	Sistem monitör ekran mesajları için sabit değer.
3-4	Değişken	Donanım Versiyonu (Major)	Donanım sürümünün 'Major' kısmını temsil eden 2 baytlık alan.
5-6	Değişken	Donanım Versiyonu (Minor)	Donanım sürümünün 'Minor' kısmını temsil eden 2 baytlık alan.
7-8	Değişken	Yazılım Versiyonu (Major)	Yazılım sürümünün 'Major' kısmını temsil eden 2 baytlık alan.
9-10	Değişken	Yazılım Versiyonu (Minor)	Yazılım sürümünün 'Minor' kısmını temsil eden 2 baytlık alan.
11-14	Değişken	Derleme Numarası (Build No)	Yazılımın derleme numarasını içeren 4 baytlık alan.
15-20	Değişken	Derleme Tarihi	Yazılımın derleme tarihini içeren 6 baytlık alan.
21	Değişken	Haberleşme Modu	Haberleşme türünü belirten bayt (ör. USB, RS485, vb.).
22	Değişken	CRC Modu	Hata kontrol modunu belirten bayt ('0x01': CRC etkin).
23	Değişken	Operasyon Modu	Sistem operasyon modunu belirten bayt (ör. Temel, Gelişmiş, vb.).
24	0x00 (Sabit değer)	Yedek Alan	Gelecekteki geliştirmeler için ayrılmıştır.
25-28	Değişken	Kart ID Değeri	Kart kimlik bilgisi için ayrılmış 4 baytlık alan.
29	Değişken	Rezerve Alan	Üretici tarafından ayrılmış, kullanmayınız.
30-31	Değişken	CRC Kontrol Değeri	CRC-16 algoritması ile hesaplanan 2 baytlık kontrol değeri.

Tablo 20 Sistem monitoring yapısının Byte Byte Açıklama Özeti

6.6.2.1.1 Örnek Sistem Monitor Mesajı Açıklaması

Bu bölümde, RFID kart okutulduğunda cihazın otomatik olarak master cihaza gönderdiği bir cevap mesajı detaylandırılmaktadır. Aşağıda verilen örnek mesaj, sistemin çalışma durumunu ve okutulan kart bilgilerini içermektedir.

İlgili Örnek Mesaj -1

AA 01 01 00 01 00 02 00 02 00 00 00 00 00 25 32 30 32 34 31 32 00 01 01 00 C3 C7 46 FC BE 24 BA

Mesaj Yapısının Byte Byte Açıklaması

Ofset (Byte)	Hex Değer	Byte Açıklaması	Açıklama
0	0xAA	Başlangıç Baytı	Mesajın başlangıcını belirler. Sabittir.
1	0x01	Cihaz Adresi	Cihazın adresini belirtir.
2	0x01	Mesaj Türü	Sistem monitör mesajı olduğunu gösterir.
3-4	0x00 0x01	Donanım Versiyonu (Major)	Cihazın ana donanım sürümü (v1.x).
5-6	0x00 0x02	Donanım Versiyonu (Minor)	Cihazın alt donanım sürümü (v1.2).
7-8	0x00 0x02	Yazılım Versiyonu (Major)	Yazılım versiyonu (v2.x).
9-10	0x00 0x00	Yazılım Versiyonu (Minor)	Yazılımın alt sürümü (v2.0).
11-14	0x00 0x00 0x00 0x25	Derleme Numarası	Yazılımın derleme numarası (37).
15-20	0x32 0x30 0x32 0x34 0x31 0x32	Derleme Tarihi	Yazılımın üretildiği tarih (2024-12).
21	0x00	Haberleşme Modu	0x00: USB haberleşme modunu gösterir (COMM_USB).
22	0x01	CRC Modu	0x01: CRC aktif, veri bütünlüğü kontrolü etkin (DATA_INTEGRITY_CHECK).
23	0x01	Operasyon Modu	0x01: Advanced modda çalışıyor (MODE_ADVANCED).
24	0x00	Yedek Alan	Gelecekteki kullanım için ayrılmıştır.
25-28	0xC3 0xC7 0x46 0xFC	Kart ID	Okutulan RFID kartın benzersiz kimlik numarası.
29	0xBE	Rezerve Alan	Üretici tarafından ayrılmış, kullanılmaz.
30-31	0x24 0xBA	CRC Kontrol Değeri	CRC-16 CCITT algoritması ile hesaplanan hata kontrol değeri.

Tablo 21 Sistem monitoring yapısı için örnek (Gerçek RFIDAX cihazı ve gerçek karttan veri alınmıştır)

Özet

Bu örnek mesaj, bir RFID kart okutulduğunda cihazın verdiği **otomatik yanıt** mesajını göstermektedir. **Mesajın toplam uzunluğu 32 byte'tır, çünkü CRC kontrolü aktiftir.** Eğer CRC kapalı olsaydı, mesajın uzunluğu **30 byte** olacaktı ve son iki CRC baytı (0x24 0xBA) yer almayacaktı.

Bu yapı, sistem monitörünün güncel durumunu ve okutulan kart bilgilerini içeren bir veri paketi oluşturarak master cihaza gönderilmesini sağlar. Kullanıcılar, bu mesajı analiz ederek cihazın durumu ve okutulan kart hakkında bilgi sahibi olabilirler.

İlgili Örnek Mesaj -2

AA 01 01 00 01 00 02 00 02 00 00 00 00 00 25 32 30 32 34 31 32 02 01 00 00 C3 C7 46 FC BE 5A BC

Ofset (Byte)	Hex Değer	Byte Açıklaması	Açıklama
0	0xAA	Başlangıç Baytı	Veri paketinin başlangıç baytı. Her zaman sabittir.
1	0x01	Cihaz Adresi	Cihaza atanmış değişken adres (1. cihaz için 0x01).
2	0x01	Mesaj Türü	Sistem monitör ekran mesajı olduğunu gösterir.
3-4	0x00 0x01	Donanım Versiyonu (Major)	Cihazın ana donanım sürümü (v1.x).
5-6	0x00 0x02	Donanım Versiyonu (Minor)	Cihazın alt donanım sürümü (v1.2).
7-8	0x00 0x02	Yazılım Versiyonu (Major)	Yazılım versiyonu (v2.x).
9-10	0x00 0x00	Yazılım Versiyonu (Minor)	Yazılımın alt sürümü (v2.0).
11-14	0x00 0x00 0x00 0x25	Derleme Numarası	Yazılımın derleme numarası (37).
15-20	0x32 0x30 0x32 0x34 0x31 0x32	Derleme Tarihi	Yazılımın üretildiği tarih (2024-12).
21	0x02	Haberleşme Modu	0x02: Type-C haberleşme modu (COMM_TYPE_C).
22	0x01	CRC Modu	0x01: CRC aktif, veri bütünlüğü kontrolü etkin (DATA_INTEGRITY_CHECK).
23	0x00	Operasyon Modu	0x00: Basic mod aktif (MODE_BASIC).
24	0x00	Yedek Alan	Gelecekteki kullanım için ayrılmıştır.
25-28	0xC3 0xC7 0x46 0xFC	Kart ID	Okutulan RFID kartın benzersiz kimlik numarası.
29	0xBE	Rezerve Alan	Üretici tarafından ayrılmış, kullanılmaz.
30-31	0x5A 0xBC	CRC Kontrol Değeri	CRC-16 CCITT hata kontrol değeri.

Tablo 22 Sistem monitoring yapısı için örnek (Gerçek RFIDAX cihazı ve gerçek karttan veri alınmıştır)

Aşağıdaki tablo, örnek 2 cihazdan gelen sistem monitoring mesajındaki **anahtar bilgileri** özetlemektedir:

Kategori	Değer
Communication Mode	Type-C
Data Integrity Check	Enabled (CRC Active)
Operation Mode	Basic
Hardware Version	1.2
Software Version	2.0
Build Number	37
Build Date	2024-12
Card ID	C3C746FC
CRC	5ABC

Bu bilgiler, cihazın haberleşme türünü, yazılım/donanım bilgilerini ve okutulan kart ID'sini içermektedir.

Mesaj 32 byte uzunluğundadır, ancak CRC devre dışı bırakıldığında mesaj 30 byte olur (son 2 byte olan CRC değeri kaldırılır).

İlgili Örnek Mesaj -3

AA 03 01 00 01 00 02 00 03 00 02 00 00 00 4B 32 30 32 35 30 31 03 01 02 00 66 A7 7B DA 60 F8 84

Ofset (Byte)	Hex Değer	Alan Adı	Detaylı Açıklama
0	0xAA	Başlangıç Baytı	Veri paketinin başlangıcını belirten sabit değer.
1	0x03	Cihaz Adresi	Cihaza atanmış adres (değişken).
2	0x01	Mesaj Türü	Sistem monitör ekran mesajları için sabit değer.
3-4	0x00 0x01	Donanım Versiyonu (Major)	Donanım sürümünün 'Major' kısmını gösteren 2 byte.
5-6	0x00 0x02	Donanım Versiyonu (Minor)	Donanım sürümünün 'Minor' kısmını gösteren 2 byte.
7-8	0x00 0x03	Yazılım Versiyonu (Major)	Yazılım sürümünün 'Major' kısmını gösteren 2 byte.
9-10	0x00 0x02	Yazılım Versiyonu (Minor)	Yazılım sürümünün 'Minor' kısmını gösteren 2 byte.
11-14	0x00 0x00 0x00 0x4B	Derleme Numarası (Build No)	Yazılımın derleme numarasını belirten 4 byte.
15-20	0x32 0x30 0x32 0x35 0x30 0x31	Derleme Tarihi	Yazılımın derleme tarihini belirten 6 byte (Yıl/Ay/Gün).
21	0x03	Haberleşme Modu	0x03 - Mobile iletişim modu seçildi (COMM_MOBILE).
22	0x01	CRC Modu	0x01 - CRC aktif, veri bütünlüğü kontrolü etkin (DATA_INTEGRITY_CHECK).
23	0x02	Operasyon Modu	0x02 - Enterprise mod aktif (MODE_ENTERPRISE).
24	0x00	Yedek Alan	Sabit değer 0x00, gelecekteki geliştirmeler için ayrılmıştır.
25-28	0x66 0xA7 0x7B 0xDA	Kart ID Değeri	Kart kimlik bilgisi için ayrılmış 4 byte.
29	0x60	Rezerve Alan	Üretici tarafından ayrılmış, kullanılmaz.
30-31	0xF8 0x84	CRC Kontrol Değeri	CRC-16/MODBUS algoritması ile hesaplanan 2 byte kontrol değeri.

Tablo 23 Sistem monitoring yapısı için örnek (Gerçek RFIDAX cihazı ve gerçek karttan veri alınmıştır)

Aşağıdaki tablo, örnek 3 cihazdan gelen sistem monitoring mesajındaki **anahtar bilgileri** özetlemektedir

Kategori	Değer
Communication Mode	Mobile (COMM_MOBILE)
Data Integrity Check	Enabled (CRC Active)
Operation Mode	Enterprise (MODE_ENTERPRISE)
Hardware Version	1.2
Software Version	3.2
Build Number	75
Build Date	2025-01
Card ID	66A77BDA
CRC	F884

6.7 Cihazdan Gelen mesaj yapısı

Bu bölüm RFIDAX cihazlarının cihazdan gelen mesaj yapısı anlatır.

Bu bölüm, **cihazın gönderdiği mesajların nasıl yapılandırıldığını ve nasıl anlamlandırılacağını** açıklamaktadır. Cihazdan iki farklı mesaj yapısı gelmektedir:

Durum Mesajları (report_status)

Cihazın çalışma durumu, hata kodları veya başarılı işlem bilgisi içeren mesaj türüdür.

Kısa ve sabit uzunluktadır (6 byte).

Hata veya başarı durumlarını haberleşme türüne ve veri bütünlüğü moduna göre raporlar.

Eğer talep edilirse (Custom Mode) bu hata modlarında açıklayıcı bir şekilde iletebilir.

Veri Paketleri (send_data_packet)

Cihazın RFID türü kartlardan okuduğu verileri, cihazın kendi kombinasyonu, cihaz ayarlarının ve cihaz hafızasında yer alan verilerin yollandığı mesaj yapısıdır.

İlgili komuta göre gönderilen cevapları iletir.

Değişken uzunluktadır.

Gönderilecek veri paketini, haberleşme türüne ve veri bütünlüğü moduna göre kullanıcı cihaza gönderir.

Bu iki mesaj yapısı aşağıdaki bölümlerde detaylıca açıklanmıştır.

6.7.1 Durum Mesajları (Report_status)

Bu bölüm, cihazın gönderdiği **durum mesajlarını** anlamlandırmak için hazırlanmıştır.

Cihaz, belirli olaylar veya hata durumları oluştuğunda, **report_status()** fonksiyonu aracılığıyla **6 byte'lık** bir mesaj oluşturur ve hedef cihaza iletir.

Mesaj, **başlık (HEADER)**, **cihaz adresi (DEVICE_ADDRESS)**, **durum kodu (STATUS_CODE)** ve opsiyonel olarak **CRC bilgilerini** içerir.

CRC etkinse, mesajın son iki byte'ı **veri bütünlüğünü sağlamak için** kullanılmaktadır.

Ek olarak, açıklamalı mesajlar için **with_description** parametresi aktif hale getirildiğinde, hata kodunun anlamını içeren bir metin de iletebilir. Bu yapı sayesinde, cihazın çalışma durumu ve olası hatalar kolayca analiz edilebilir. Bu **with_description** konusu bu dokümanda açıklanmayacaktır.

Bu fonksiyon, **cihazın durumunu, yapılan işlemin durumunu veya hata kodlarını bildirmek için kullanılan mesaj formatını** oluşturur. Mesaj, cihazın hata veya durum bilgilerini hedef cihaza iletmek amacıyla belirli bir yapıdadır. Bu yapı aşağıdadır.

HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
--------	----------------	------------------	-----------------	-------------------------	------------------------

Header mesaj bilgisi her zaman report status sabittir. Her zaman hex olarak 0x BB'dir.

Device adres sonradan değişebilir, fabrika çıkışlı cihazlarda hex değeri 0x01'dir.

STATUS_CODE_HIGH ve STATUS_CODE_LOW değerleri gelen duruma göre değişiklik gösterir.

Bu değerlerin tam listesi için bölüm [6.4 Cihaz Durum Mesajı, Hata Kodları ve Açıklamaları](#) kısmına bakınız.

Eğer Crc aktif ise ilgili haberleşme protokolün Crc değerleri sona eklenir.

Ek olarak, açıklamalı mesajlar için **with_description** parametresi aktif hale getirildiğinde, hata kodunun anlamını içeren bir metin de iletebilir. Bu yapı sayesinde, cihazın çalışma durumu ve olası hatalar kolayca analiz edilebilir.

Durum Mesajları (Report_status) Örneği 1

Byte	İçerik	Açıklama	Örnek Değer
0	HEADER	Mesaj başlangıç baytı	0xBB
1	DEVICE_ADDRESS	Cihaz adresi	0x01
2	STATUS_CODE_HIGH	Durum kodunun yüksek baytı	0x00
3	STATUS_CODE_LOW	Durum kodunun düşük baytı	0x00
4	CRC_HIGH (opsiyonel)	CRC yüksek baytı (Veri bütünlüğü için)	0x5C
5	CRC_LOW (opsiyonel)	CRC düşük baytı (Veri bütünlüğü için)	0x3E

Tablo 24 (Gerçek RFIDAX cihazı ve gerçek karttan veri alınmıştır)

BB 01 00 00 5C 3E incelendiğinde 2. Byte 0x00 ve 3 byte 0x00 değeri” 0x0000 SUCCESS – İşlem başarılı.” Karşılık gelmektedir. Lütfen ilgili bölümü inceleyiniz. ([6.4.1 Genel Sistem Durumu ve Hataları](#))

Durum Mesajları (Report_status) Örneği 2

Durum mesajını daha anlaşılabilirliği için hatalı led komutu gönderilmiştir.

Gönderilen Mesaj

Command Sent						
0xAA	0x01	0x01	0x01	0x03	0x3F	0x10

Gelen mesaj aşağıdadır,

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xBB	0x01	0x00	0x10	0x4E	0x0F

BB 01 00 10 4E 0F incelendiğinde 2. Byte 0x00 ve 3 byte 0x10 değeri” ERR_INVALID_LED_STATE – Geçersiz LED durumu. LED’in açık veya kapalı olup olmadığını doğru kontrol edin.” Karşılık gelmektedir. Lütfen ilgili bölümü inceleyiniz. ([6.4.3 LED ve Göstergeler ile İlgili Hatalar](#))



Advanced RFID Card Reader Writer

Durum Mesajları (Report_status) Örneği 3

Durum mesajını daha anlaşılabilirliği için buzzer'ı 100ms aktif et komutu gönderilmiştir.

Gönderilen Mesaj

Command Sent						
0xAA	0x01	0x06	0x00	0x0A	0x18	0x98

Gelen mesaj aşağıdadır,

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xBB	0x01	0x00	0x00	0x5C	0x3E

BB 01 00 BB 01 00 00 5C 3E incelendiğinde 2. Byte 0x00 ve 3 byte 0x00 değeri" 0x0000 SUCCESS – İşlem başarılı." Karşılık gelmektedir. Lütfen ilgili bölümü inceleyiniz. ([6.4.1 Genel Sistem Durumu ve Hataları](#))

Buzzer 100ms aktif edilme isteğine başarılı bir mesaj döndürüp buzzer 100ms açmıştır.

Durum Mesajları (Report_status) Örneği 4

Durum mesajını daha anlaşılabilirliği RFID karttan 2 bloğu okuma isteği gönderilmiştir.

Cihazın anten alanına okuması için kart eklenmemiştir.

Gönderilen Mesaj

Command Sent								
0xAA	0x01	0x08	0x01	0x02	0x02	0x02	0xDE	0xF6

Gelen mesaj aşağıdadır,

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xBB	0x01	0x00	0x20	0x78	0x5C

BB 01 00 10 4E 0F incelendiğinde 2. Byte 0x00 ve 3 byte 0x20 değeri" ERR_CARD_NOT_FOUND – Kart bulunamadı. Kartın okuyucu alanına uygun şekilde yerleştirildiğini kontrol edin." Karşılık gelmektedir. Lütfen ilgili bölümü inceleyiniz. ([6.4.4 RFID Kart Okuma/Yazma Hataları](#))



Advanced RFID Card Reader Writer

Durum Mesajları (Report_status) Örneği 5

Durum mesajını daha anlaşılabilirliği için RFID karttan 2 bloğu okuma isteği gönderilmiştir.

Üsteki örnekten farkı ise CRC'nin aktif olmadığı cihaz ([Data Integrity Check]: Disabled (No CRC)) için farkını göstermek için eklenmiştir.

Cihazın anten alanına okuması için kart eklenmemiştir.

Gönderilen Mesaj

Command Sent								
0xAA	0x01	0x08	0x01	0x02	0x02	0x02	0xDE	0xF6

Gelen mesaj aşağıdadır,

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xBB	0x01	0x00	0x20	0x00	0x00

BB 01 00 20 4E 0F incelendiğinde 2. Byte 0x00 ve 3 byte 0x20 değeri” ERR_CARD_NOT_FOUND – Kart bulunamadı.

Kartın okuyucu alanına uygun şekilde yerleştirildiğini kontrol edin.” Karşılık gelmektedir. Lütfen ilgili bölümü inceleyiniz. ([6.4.4 RFID Kart Okuma/Yazma Hataları](#))

Dikkat edilirse CRC_HIGH ve CRC_LOW byte 0x00 gelmiştir, bu Veri bütünlüğü modunun NO_DATA_INTEGRITY_CHECK olduğunu gösterir.

6.7.2 Veri Paketleri (send_data_packet)

Bu bölüm, cihazın gönderdiği **veri paketlerini** anlamlandırmak için hazırlanmıştır. Cihaz, belirli işlemler sonucunda veya bir isteğe yanıt olarak **send_data_packet ()** fonksiyonu aracılığıyla **dinamik uzunlukta bir mesaj** oluşturur ve hedef cihaza iletir.

Mesaj, **başlık (HEADER)**, **cihaz adresi (DEVICE_ADDRESS)**, **mesaj türü (MESSAGE_TYPE)**, **veri (DATA)** ve opsiyonel olarak **CRC bilgilerini** içerir. CRC etkinse, mesajın son iki byte'ı **veri bütünlüğünü sağlamak için** kullanılmaktadır. Bu yapı sayesinde, cihazın gönderdiği veriler standart bir formatta işlenebilir ve hedef cihaz tarafından doğru şekilde alınabilir.

HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xAA	0x00-0xFF 1 Byte	Değişken 1 Byte	Değişken Uzunluk	Değişken 1 Byte	Değişken 1 Byte

Bu yapı, cihazın gönderdiği veri paketlerinin standart formatını tanımlar ve mesajın farklı bileşenlerini açıklar.

HEADER (0xAA), Mesaj başlangıç baytıdır ve her zaman sabittir.

Cihaz, send_data_packet fonksiyonunu kullanarak veri gönderdiğinde, mesajın ilk byte'ı her zaman 0xAA olacaktır.

DEVICE_ADDRESS (0x01 -255 Değişken), Cihaz adresidir ve fabrika çıkışlı cihazlarda 0x01 olarak ayarlanmıştır.

İhtiyaca göre değiştirilebilir.

MESSAGE_TYPE (Farklı Değerler Alabilir), Cevap verilen fonksiyonun türünü belirtir.

Her fonksiyonun kendine ait bir Message Type değeri bulunmaktadır.

Bu değerler için [6.7.2.1 Message Type Bilgisi](#) bölümüne bakınız.

DATA (5 Byte- 1024 Byte Arası Değişken Uzunluk), Gönderilen veri içeriğini içerir.

Uzunluğu 5 byte'tan 1024 byte'a kadar değişebilir.

CRC_HIGH & CRC_LOW (Opsiyonel), STATUS_CODE_HIGH ve STATUS_CODE_LOW değerleri gelen duruma göre değişiklik gösterir. Her mesaj için ayrı bir CRC hesaplanır. Eğer CRC aktifse, kullanılan haberleşme protokolüne göre hesaplanarak mesajın sonuna eklenir.

6.7.2.1 Message Type Bilgisi

Message type bilgisi send data paketinin bir byte değeridir. Bu bilginin amacı RFIDAX cihazının gönderdiği cevabın türünü ve fonksiyonunu belirlemektir.

Mesaj type bilgisine göre hangi işlem ve fonksiyon cevabı belirtildiğini anlaşılmaktadır.

1 byte'lık veridir.

Send data paketinden gelen mesajın yorumlanması büyük kolaylık sağlamaktadır.

Message Type bilgisi ve açıklamaları aşağıdadır.

Cihazdan gelen mesaj tipleri, sistemin işleyişini anlamak ve verileri doğru şekilde yönetmek için kritik bir öneme sahiptir. Her mesaj tipi, belirli bir işlem veya veriyi temsil eder ve cihazın çalışma mantığını organize eden temel yapı taşlarından biridir.

Mesaj Kodu	Mesaj Tipi	Açıklama
0x01	System_Monitor	Sistem monitör verisini göndermek için kullanılan mesaj tipi.
0x02	Device_Version_Info	Cihaz versiyonu ile ilgili tüm veriyi göndermek için kullanılan mesaj tipi.

Tablo 25 Message Type Bilgisi

6.7.2.1.1 RFID Kart Okuma İşlemleri Mesajları ve Message Type Bilgisi

Cihaz, okunan RFID kartların UID, SAK, ATQA gibi kritik bilgilerini bu mesaj tipleri aracılığıyla iletir. Aynı zamanda, kartlara özel anahtarlar (KeyA, KeyB'i) ve opsiyonel güvenlik anahtarları da mesajlarla paylaşılabilir.

Mesaj Kodu	Mesaj Tipi	Açıklama
0x07	Read_Card_UID	Okunan RFID kartın UID bilgisini içeren mesaj tipi.
0x08	Read_Card_ATQA	Okunan RFID kartın ATQA bilgisini içeren mesaj tipi.
0x09	Read_Card_SAK	Okunan RFID kartın SAK bilgisini içeren mesaj tipi.
0x0A	Read_Card_Info	Okunan RFID kartın UID, SAK ve ATQA değerlerini tek mesajda gönderen mesaj tipi.
0x0B	Display_Key_A	RFID kartlar için keyA değeri (0x11 aynıdır, kullanılma amacı veri bütünlüğüdür)
0x0C	Display_Key_B	RFID kartlar için keyB değeri (0x12 aynıdır, kullanılma amacı veri bütünlüğüdür)
0x0E	Display_Key_Optional_KeyA	RFID kartlar için opsiyonel KeyA değerini gönderen mesaj tipi.
0x0F	Display_Key_Optional_KeyB	RFID kartlar için opsiyonel KeyB değerini gönderen mesaj tipi.
0x10	Blok_Verilerini_Okuma	RFID kartın belirli bir bloğunu okumak için kullanılan mesaj tipi.
0x11	KeyA	RFID kart için KeyA değerini içeren mesaj tipi.
0x12	KeyB	RFID kart için KeyB değerini içeren mesaj tipi.

Tablo 26 RFID Kart Okuma İşlemleri Mesajları ve Message Type Bilgisi

6.7.2.1.2 Flash Bellek ID Güncelleme Mesajları ve Message Type Bilgisi

Cihazın iç belleğinde depolanan kimlik ve yapılandırma verileri, zaman zaman güncellenmesi gereken kritik bilgileri içerir. Bu mesaj tipleri, bellek üzerindeki belirli ID'leri güncellemek için kullanılır.

Mesaj Kodu	Mesaj Tipi	Açıklama
0x20	Flash_Update_ID_0	Flash bellekteki ID 0'ı güncellemek için kullanılan mesaj tipi.
0x21	Flash_Update_ID_1	Flash bellekteki ID 1'i güncellemek için kullanılan mesaj tipi.
0x22	Flash_Update_ID_2	Flash bellekteki ID 2'yi güncellemek için kullanılan mesaj tipi.
0x23	Flash_Update_ID_3	Flash bellekteki ID 3'ü güncellemek için kullanılan mesaj tipi.
0x24	Flash_Update_ID_4	Flash bellekteki ID 4'ü güncellemek için kullanılan mesaj tipi.
0x25	Flash_Update_ID_5	Flash bellekteki ID 5'i güncellemek için kullanılan mesaj tipi.
0x26	Flash_Update_ID_6	Flash bellekteki ID 6'yı güncellemek için kullanılan mesaj tipi.
0x27	Flash_Update_ID_7	Flash bellekteki ID 7'yi güncellemek için kullanılan mesaj tipi.
0x28	Flash_Update_ID_8	Flash bellekteki ID 8'i güncellemek için kullanılan mesaj tipi.
0x29	Flash_Update_ID_9	Flash bellekteki ID 9'u güncellemek için kullanılan mesaj tipi.

Tablo 27 Flash Bellek Güncelleme Mesajları ve Message Type Bilgisi

6.7.2.1.3 Flash Bellekten ID Okuma Mesajları ve Message Type Bilgisi

Cihazın iç belleğinde depolanan kimlik ve yapılandırma verileri, zaman zaman güncellenmesi gereken kritik bilgileri içerir. Bu mesaj tipleri, bellek üzerindeki belirli ID'leri okumak için kullanılır.

Mesaj Kodu	Mesaj Tipi	Açıklama
0x50	Flash_Read_ID_0	Flash bellekten ID 0'ı okumak için kullanılan mesaj tipi.
0x51	Flash_Read_ID_1	Flash bellekten ID 1'i okumak için kullanılan mesaj tipi.
0x52	Flash_Read_ID_2	Flash bellekten ID 2'yi okumak için kullanılan mesaj tipi.
0x53	Flash_Read_ID_3	Flash bellekten ID 3'ü okumak için kullanılan mesaj tipi.
0x54	Flash_Read_ID_4	Flash bellekten ID 4'ü okumak için kullanılan mesaj tipi.
0x55	Flash_Read_ID_5	Flash bellekten ID 5'i okumak için kullanılan mesaj tipi.
0x56	Flash_Read_ID_6	Flash bellekten ID 6'yı okumak için kullanılan mesaj tipi.
0x57	Flash_Read_ID_7	Flash bellekten ID 7'yi okumak için kullanılan mesaj tipi.
0x58	Flash_Read_ID_8	Flash bellekten ID 8'i okumak için kullanılan mesaj tipi.
0x59	Flash_Read_ID_9	Flash bellekten ID 9'u okumak için kullanılan mesaj tipi.

Tablo 28 Flash Bellekten Okuma Mesajları ve Message Type Bilgisi

6.7.2.1.4 Flash Bellek Sıfırlama Mesajları ve Message Type Bilgisi

Bazen cihazın belleğinde bulunan eski verilerin temizlenmesi veya belirli kimlik bilgilerinin sıfırlanması gerekebilir. Bu mesaj tipleri, belirli ID'leri sıfırlamak için kullanılır.

Mesaj Kodu	Mesaj Tipi	Açıklama
0x8C	Flash_Reset_ID_0	Flash bellekteki ID 0'ı sıfırlamak için kullanılan mesaj tipi.
0x8D	Flash_Reset_ID_1	Flash bellekteki ID 1'i sıfırlamak için kullanılan mesaj tipi.
0x8E	Flash_Reset_ID_2	Flash bellekteki ID 2'yi sıfırlamak için kullanılan mesaj tipi.
0x8F	Flash_Reset_ID_3	Flash bellekteki ID 3'ü sıfırlamak için kullanılan mesaj tipi.
0x90	Flash_Reset_ID_4	Flash bellekteki ID 4'ü sıfırlamak için kullanılan mesaj tipi.
0x91	Flash_Reset_ID_5	Flash bellekteki ID 5'i sıfırlamak için kullanılan mesaj tipi.
0x92	Flash_Reset_ID_6	Flash bellekteki ID 6'yi sıfırlamak için kullanılan mesaj tipi.
0x93	Flash_Reset_ID_7	Flash bellekteki ID 7'yi sıfırlamak için kullanılan mesaj tipi.
0x94	Flash_Reset_ID_8	Flash bellekteki ID 8'i sıfırlamak için kullanılan mesaj tipi.
0x95	Flash_Reset_ID_9	Flash bellekteki ID 9'u sıfırlamak için kullanılan mesaj tipi.
0xC8	Flash_Display_All_IDs	Flash bellekteki tüm ID'leri görüntülemek için kullanılan mesaj tipi.

Tablo 29 Flash Bellek Sıfırlama Mesajları ve Message Type Bilgisi

Sıfırlanana ID yerine sistem otomatik olarak 0xFF verisi yerleştirilir.

6.7.2.1.5 Format ID Okuma Mesajı ve Message Type Bilgisi

Cihazın tanımlayıcı format bilgisi, sistemin veri işleme süreçlerinde kritik bir rol oynar. Bu mesaj, format ID'sinin kimlik numarasını okumak ve doğrulamak için kullanılır.

Mesaj Kodu	Mesaj Tipi	Açıklama
0x60	ReadFlash_Format_ID_DATA	Format ID'sinin kimlik numarasını okumak için kullanılan mesaj tipi.

Tablo 30 Flash Bellek Sıfırlama Mesajları ve Message Type Bilgisi

6.7.2.1.6 Sector Trailer İşlemleri ve Message Type Bilgisi

RFID kartların her sektöründe yer alan sector trailer blokları, erişim kontrol yapısını belirler. Bu yapı; Key A, erişim bitleri ve Key B olmak üzere üç temel alandan oluşur. Aşağıdaki mesaj tipi, cihazın bu yapıyı formatlamasını sağlar.

Mesaj Kodu	Mesaj Tipi	Açıklama
0x13	Format_Sector_Trailer	Belirli bir sektörün trailer bloğunu KeyA, KeyB ve erişim bitleri ile yapılandırmak için kullanılır.

Tablo 31-A Sector Trailer Formatlama Mesajı ve Message Type Bilgisi

Not: Bu işlem sırasında, cihazda tanımlı optionalKeyA ve optionalKeyB anahtarları kullanılır. Erişim bitleri sabit olarak 7F 07 88 40 şeklinde ayarlanır. Mesajla birlikte toplam 16 baytlık bir sector trailer verisi gönderilir.



Advanced RFID Card Reader Writer

Veri Paketleri (send_data_packet) Örneği 1

Veri paketi(send_data_packet) daha anlaşılabilirliği 2 numaralı ID güncelleme yapılacaktır.

Gönderilen Mesaj

Command Sent											
0xAA	0x01	0x0C	0x01	0x04	0x02	0xAA	0x22	0xCC	0xDD	0x8B	0x84

Gelen mesaj aşağıdadır,

Send Data Packet Response								
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA				CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xAA	0x01	0x22	0xAA	0x22	0xCC	0xDD	0xED	0x8A

AA 01 22 AA 22 CC DD ED 8A incelendiğinde 2 byte bilgisi Message Type Bilgisidir, değeri 0x22'dir.

Bunun *Message Type Bilgisi* tablosundan([6.7.2.1.2 Flash Bellek ID Güncelleme Mesajları ve Message Type Bilgisi](#)), karşılığına baktığımızda "Flash_Update_ID_2 Flash bellekteki ID 2'yi güncellemek için kullanılan mesaj tipi." bilgisini verir.

3,4,5,6. Byte bilgisinde data mesajdır, güncellenen ID bilgisini verir. Sırayla bu bilgiler 0xAA,0x22,0xCC,0xDD'dir.

Dikkat edilirse CRC_HIGH ve CRC_LOW byte bilgileri 0xED ve 0x8A gelmiştir, bu Veri bütünlüğü modunun

DATA INTEGRITY CHECK olduğunu gösterir.

Veri Paketleri (send_data_packet) Örneği 2

Veri paketi(send_data_packet) daha anlaşılabilirliği UID read(kart id okuma) işlemi yapılacaktır.

Gönderilen Mesaj

Command Sent						
0xAA	0x01	0x07	0x01	0xFF	0xA3	0x23

Gelen mesaj aşağıdadır,

Send Data Packet Response								
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA				CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xAA	0x01	0x07	0x66	0xA7	0x7B	0xDA	0x1D	0x28

AA 01 07 66 A7 7B DA 1D 28 incelendiğinde 2 byte bilgisi Message Type Bilgisidir, değeri 0x07'dir.

Bunun *Message Type Bilgisi* tablosundan ([6.7.2.1.1 RFID Kart Okuma İşlemleri Mesajları ve Message Type Bilgisi](#)), karşılığına baktığımızda "Okunan RFID kartın UID bilgisini içeren mesaj tipi." bilgisini verir.

3,4,5,6. Byte bilgisinde data mesajdır, güncellenen ID bilgisini verir. Sırayla bu bilgiler 0x66,0xA7,0x7B,0xDA'dir.

Dikkat edilirse CRC_HIGH ve CRC_LOW byte bilgileri 0x1D ve 0x28 gelmiştir, bu Veri bütünlüğü modunun

DATA INTEGRITY CHECK olduğunu gösterir.

Veri Paketleri (send_data_packet) Örneği 3

Veri paketi(send_data_packet) daha anlaşılabilmesi Device Info bilgisi sorgu yapılacaktır.

Gönderilen Mesaj

Command Sent					
0xAA	0x01	0x0D	0x01	0x57	0x50

Gelen mesaj aşağıdadır, 00 01 00 02 00 03 00 02 00 00 00 4B 32 30 32 35 30 31 03 01 01 00

Send Data Packet Response						
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)	
0xAA	0x01	0x02	0x00 0x01 0x00 0x02 0x00 0x03 0x00 0x02 0x00 0x00 0x00 0x4B 0x32 0x30 0x32 0x35 0x30 0x31 0x03 0x01 0x01 0x00	0xF7	0xEC	

AA 01 02 00 01 00 02 00 03 00 02 00 00 00 4B 32 30 32 35 30 31 03 01 01 00 F7 EC incelendiğinde 2 byte bilgisi Message Type Bilgisidir, değeri 0x02'dir.

Bunun *Message Type Bilgisi* tablosundan([6.7.2.1 Message Type Bilgisi](#)), karşılığına baktığımızda "Device_Version_Info Cihaz versiyonu ile ilgili tüm veriyi göndermek için kullanılan mesaj tipi." bilgisini verir.

3-24 Byte bilgisinde data mesajıdır versiyon mesajı bilgisini verir. Bu data yapısı detaylı incelemesi için..... bakınız.

Veri Paketleri (send_data_packet) Örneği 4

Veri paketi(send_data_packet) daha anlaşılabilmesi Device Format ID sorgusu yapılacaktır.

Gönderilen Mesaj

Command Sent					
0xAA	0x01	0x0F	0x02	0x01	0x51

Gelen mesaj aşağıdadır,

Send Data Packet Response								
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)			
0xAA	0x01	0x60	0x93 0x32 0xEF	0xF6	0xE2	0x20		

AA 01 60 93 32 EF F6 E2 20 incelendiğinde 2 byte bilgisi Message Type Bilgisidir, değeri 0x60'tır.

Bunun *Message Type Bilgisi* tablosundan([Format ID Okuma Mesajı ve Message Type Bilgisi](#)), karşılığına baktığımızda "ReadFlash_Format_ID_DATA Format ID'sinin kimlik numarasını okumak için kullanılan mesaj tipi.." bilgisini verir.

3,4,5,6. Byte bilgisinde data mesajıdır, sistemi formatlayan ID bilgisini verir. Sırayla bu bilgiler 0x93 ,0x32 ,0xEF ,0xF6'dır.Dikkat edilirse CRC_HIGH ve CRC_LOW byte bilgileri 0xE2 ve 0x20 gelmiştir.



Advanced RFID Card Reader Writer

7.0 Cihaz Komutları Kılavuzu

Bu kılavuz, cihazın desteklediği **komut setlerini** ve her bir komutun **işlevini**, **kullanım amacını** ve **çalışma prensibini** açıklamaktadır. Komutlar, cihazın **kontrol edilmesini**, **veri alışverişini sağlamayı** ve **belirli işlemleri gerçekleştirmeyi** mümkün kılar.

Cihaz komutları belirli bir **başlık (HEADER)** ile başlar ve **cihaz adresi**, **komut tipi**, **parametreler** ve **opsiyonel olarak CRC bilgilerini** içerir, **dinamik uzunlukta bir mesajdır**ve hedef RFIDAX bu komutlarla kontrol edilir.

HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	DATA	CRC (opsiyonel)
--------	----------------	--------------	------------------	------	--------------------

Bölüm	Açıklama
HEADER	Komutun başlangıcını belirler. Sabit bir değer alır ve tüm mesajların başında bulunur.
DEVICE_ADDRESS	Hangi cihaza komutun gönderileceğini belirler. Cihazın adresini içerir.
COMMAND_TYPE	Komutun ana türünü belirleyen anahtar koddur. Örneğin, LED kontrolü , RFID okuma , bellek işlemleri gibi farklı komut türlerini ifade eder.
SUB_TYPE	Bazı komutlarda, alt işlem türünü belirlemek için kullanılır . Örneğin, RFID komutlarında UID , SAK veya ATQA okuma seçenekleri gibi farklı alt komutları ifade eder.
DATA (Varsa)	Komutun içeriğine bağlı ek verileri içerir. Örneğin, Bir RFID karta belirli bloğuna bir veri yazılacaksa burada yer alır.
CRC (Opsiyonel)	Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır . Eğer aktifse, mesajın sonuna eklenir ve bütünlüğü doğrulamak için kullanılır.

Tablo 32 Komut Bileşenleri ve Açıklamaları

Command_Type bilgisi ilgili komutların anlatımında sıra ile açıklanacaktır.

7.1 Komut Kategorileri ve Açıklamaları

Cihazın komutları farklı kategorilere ayrılmıştır:

Sistem Kontrol Komutları (Cihazın genel kontrolü ve yönetimi için kullanılır.)

LED ve Buzzer Komutları (Görsel ve sesli uyarılar için kullanılır.)

RFID Kart Komutları (Kart okuma, yazma ve doğrulama işlemleri için kullanılır.)

Bellek Yönetim Komutları (Flash bellekte okuma, yazma ve sıfırlama işlemleri için kullanılır.)

Cihaz Yapılandırma Komutları (Baud rate, haberleşme protokolü, cihaz ID'si gibi sistem ayarlarını değiştirmek için kullanılır.)

Fabrika Ayarlarına Dönüş Komutları (Cihazı sıfırlamak ve fabrika ayarlarına döndürmek için kullanılır.)

Sector Trailer Yönetim Komutları (RFID karttaki sector trailer bloklarını oluşturmak, yalnızca trailer bloklarını güncellemek, kartın tüm bloklarını sıfırlamak veya kartı tamamen formatlamak için kullanılır.)

Bu bölümle birlikte cihaza **4 yeni alt komut** eklenmiş olup, cihaz artık toplamda **35 komut ve alt komutu** desteklemektedir.

7.1.1 Komut Kategorileri ve Alt Katagorilerin Çalışma Modlara Açıklamaları

Bu bölümde, RFIDAX cihazına gönderilebilen komutların, farklı çalışma modlarında nasıl desteklendiği açıklanmaktadır. Cihazın işlevselliği **MODE_BASIC**, **MODE_ENTERPRISE** ve **MODE_ADVANCED** olmak üzere üç farklı modda yapılandırılabilir. Bu modlar, **sistem kontrolü**, **LED ve buzzer yönetimi**, **RFID kart işlemleri**, **bellek yönetimi**, **cihaz yapılandırması**, **fabrika ayarlarına dönüş** ve **sector trailer yönetimi** gibi temel kategorilerde gruplanmıştır. Aşağıdaki tablo, **hangi komutların hangi modlarda aktif olduğunu** göstererek, kullanıcıların seçtikleri çalışma moduna göre hangi özelliklerden yararlanabileceklerini net bir şekilde anlamalarını sağlar.

MODE_BASIC, temel işlevleri içeren giriş seviyesi bir kullanım sunarken,

MODE_ENTERPRISE, gelişmiş güvenlik ve yönetim seçenekleriyle daha geniş kapsamlı işlemler yapmaya olanak tanır.

MODE_ADVANCED ise cihazın tüm yeteneklerini açarak, **maksimum özelleştirme ve yönetim imkânı** sunan en kapsamlı moddur. Bu yapı sayesinde kullanıcılar, **ihtiyaçlarına ve projelerinin gereksinimlerine** en uygun çalışma modunu seçerek cihazdan **en verimli şekilde yararlanabilirler**.

Aşağıdaki tabloda, **her bir komut kategorisinin farklı modlarda nasıl çalıştığı** ayrıntılı olarak sunulmuştur.

Kategoriler ve Komut Kodları	Mod Bilgisi			Açıklama
	MODE BASIC	MODE ENTERPRISE	MODE ADVANCED	
1- Sistem Kontrol Komutları				
0x0A	✗ (aktif değildir)	✓ (Aktiftir)	✓ (Aktiftir)	Yazılım sıfırlama komutu
0x0D	✓ (Aktiftir)	✓ (Aktiftir)	✓ (Aktiftir)	Cihaz versiyon bilgisini al
2- LED ve Buzzer Kontrol Komutları				
0x01	✓ (Aktiftir)	✓ (Aktiftir)	✓ (Aktiftir)	Belirli bir LED'i aç/kapat
0x02	✗ (aktif değildir)	✓ (Aktiftir)	✓ (Aktiftir)	Belirli bir LED'i toggle (yanıp sönme) yap
0x03	✗ (aktif değildir)	✓ (Aktiftir)	✓ (Aktiftir)	İki led sırayla LED'i toggle (yanıp sönme) yap
0x04	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	Tüm LED'leri kapat
0x05	✓ (Aktiftir)	✓ (Aktiftir)	✓ (Aktiftir)	- Buzzer'ı aç/kapat
0x06	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	Belirli süreyle buzzer çaldır
3- RFID Kart İşlem Komutları				
0x07 - RFID kart tanıma,kart okuma ve yazma komutu				
3-a) 0x07 / 0x01	✓ (Aktiftir)	✓ (Aktiftir)	✓ (Aktiftir)	Kart ID'si (UID) okuma
3-b) 0x07 / 0x02	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	SAK değeri okuma
3-c) 0x07 / 0x03	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	ATQA değeri okuma
3-d) 0x07 / 0x04	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	Kart bilgilerini okuma (UID + SAK + ATQA)
3-e) 0x07 / 0x05	✗ (aktif değildir)	✓ (Aktiftir)	✓ (Aktiftir)	Anahtarları gösterme komutu
0x08	✗ (aktif değildir)	✓ (Aktiftir)	✓ (Aktiftir)	RFID blok okuma komutu
0x09	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	RFID blok yazma komutu
4- Bellek Yönetim Komutları				
0x0B - optionalKeyA ve optionalKeyB değerlerini yazma veya okuma				
4-a) 0x0B / 0x01	✗ (aktif değildir)	✓ (Aktiftir)	✓ (Aktiftir)	keyA yazma işlemi
4-b) 0x0B / 0x02	✗ (aktif değildir)	✓ (Aktiftir)	✓ (Aktiftir)	keyB yazma işlemi
4-c) 0x0B / 0x03	✗ (aktif değildir)	✓ (Aktiftir)	✓ (Aktiftir)	keyA okuma işlemi
4-d) 0x0B / 0x04	✗ (aktif değildir)	✓ (Aktiftir)	✓ (Aktiftir)	keyB okuma işlemi

0x0C - ID işlemleri				
4-e) 0x0C / 0x01	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	ID güncelleme işlemi
4-f) 0x0C / 0x02	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	ID okuma işlemi
4-g) 0x0C / 0x03	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	ID sıfırlama işlemi
4-h) 0x0C / 0x04	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	Tüm ID'leri gösterme işlemi
5- Cihaz Yapılandırma Komutları				
0x0E - RFID yapılandırma komutları				
5-a) 0x0E / 0x01	✗ (aktif değildir)	✓ (Aktiftir)	✓ (Aktiftir)	Device ID değiştirme
5-b) 0x0E / 0x02	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	Protokol değiştirme
5-c) 0x0E / 0x03	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	Baud rate değiştirme
6- Fabrika Ayarlarına Dönüş Komutları				
0x0F - Reset Factory komutları				
6-a) 0x0F / 0x01	✗ (aktif değildir)	✓ (Aktiftir)	✓ (Aktiftir)	Tüm ayarları fabrika varsayılanlarına sıfırla
6-b) 0x0F / 0x02	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	Format RFID kartının ID'sini okuma
6-c) 0x0F / 0x03	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	Format flag set/reset etme
6-d) 0x0F / 0x04	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	Manuel olarak Format ID'yi Flash belleğe yaz
7- Sector Trailer Yönetim Komutları				
0x11 - Sector Trailer ve Format Komutları				
7-a) 0x11 / 0x01	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	Sector trailer veri yapısını oluştur (KeyA, KeyB, Access Bits 7F 07 88 40)
7-b) 0x11 / 0x02	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	Yalnızca sector trailer bloklarını özel Key'lerle güncelle
7-c) 0x11 / 0x03	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	Tüm veri bloklarını sıfırla, sector trailer bloklarını yeniden yapılandır
7-d) 0x11 / 0x04	✗ (aktif değildir)	✗ (aktif değildir)	✓ (Aktiftir)	RFID kartı tamamen formatla (blokları sil ve trailer'ları güncelle)

Tablo 33 Komut Kategorileri Ve Alt Kategorilerin Çalışma Modlara Açıklamaları



7.2 LED ve Buzzer Kontrol Komutları

LED ve buzzer kontrol komutları, sistemdeki ışıklı ve sesli uyarı mekanizmalarını yönetmek için kullanılır. Bu komutlar, belirli LED'leri açma, kapatma veya yanıp sönmesini sağlama gibi işlevler sunarken, buzzer için de açma, kapatma veya belirli süreyle ses çıkarmayı mümkün kılar. Kullanıcı, sistemin ihtiyacına göre uygun komutu seçerek LED'ler ve buzzer üzerinde tam kontrol sağlayabilir.

7.2.1 Cihazdaki Ledlerin Kontrol Komutları

Cihazdaki ledleri tam anlamıyla kontrol edebilmek için led numarası ve led kontrol komutları incelenmesi gerekmektedir.

Cihazda iki farklı led vardır. Bu ledlerin renkleri ise kullanılan cihaz versiyon bilgisine göre değişmektedir.

Donanım versiyonlarına göre led renk bilgisi Donanım Major bilgisi 1.0 donanım versiyonu minor 1.1 üstü cihazlar için kırmızı-mavidir, daha düşük versiyonlarda ise bu renkler kırmızı-sarı renklidir.

Bu Komut yapısında data verisine gerek yoktur. COMMAND_TYPE, SUB_COMMAND_TYPE verileri ile cihaz led kontrolü sağlanır.

7.2.1.1 Cihazdaki Ledlerin Led Aç/Kapat Komutları

Bu komut yapısında cihazın üzerindeki led renklerine göre ledleri açma ve kapatma işlemi yapılmaktadır.

Gönderilen mesajın 3. Byte Hex bilgisi led AÇ/KAPAT komut bilgisini verir 4 numaralı Hex bilgisi led numarası bilgisini verir, 5 byte numaralı hex bilgisi ise AÇMA/KAPATMA bilgisini verir.

4 numaralı byte bilgisi 0x01 veya 0x03 olabilir, bunun haricinde cihaz hata bilgisi olarak ERR_LED_NUMBER_INVALID bilgisi, bu hata bilgisi için lütfen "[6.4.3 LED ve Göstergeler ile İlgili Hatalar](#)" bakınız.

Hex bilgisi 0x01 numaralı ledin rengi kırmızıdır, hex bilgisi 0x03 numaralı ledin rengi versiyon bilgisine göre MAVİ veya SARI'dır.

5.byte bilgisi ise AÇMA/KAPATMA bilgisini verir, 0x00 veya 0x01 olabilir harici bir durumda ise

ERR_INVALID_LED_STATE hatasını verir, bu hata bilgisi için lütfen "[6.4.3 LED ve Göstergeler ile İlgili Hatalar](#)" bakınız.

Cihazdaki ledlerin kontrol komutları aşağıdadır,

7.2.1.1.1 Mavi veya Sarı Ledin Kontrol Komutları

Gönderilen Mesaj

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	DATA	CRC (opsiyonel)
0xAA	0x01	0x01	0x03 0x01	NONE	0x79 0x30

HEADER **0xAA** Komutun başlangıcını belirler. Sabit bir değer alır ve tüm mesajların başında bulunur.

DEVICE_ADDRESS **0x01** Cihazın adresini içerir.

COMMAND_TYPE **0x01** Komutun ana türünü belirleyen anahtar koddur, Ledlerin Led Aç/Kapat komutudur.

SUB_TYPE 0x03 led numarasıdır (mavi led), 0x01 ise led aç komudur. Eğer ilgili ledi kapatmak istenirse 5. Byte hex verisine 0x00 gönderilmez. (Bu durumda ilgili CRC değerinin değiştiğini unutmayın.)

CRC (Opsiyonel) CRC dönen hex değeri **0x79 0x30**.

Gelen mesaj aşağıdadır,

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xBB	0x01	0x00	0x00	0x5C	0x3E

Gönderilen led komuduna göre dönen mesaj başarılı bilgisi içermektedir. Bu mesaj bilgisi “STATUS_CODE_HIGH Ve STATUS_CODE_LOW “anlaşılmaktadır. Genel Sistem Durumu ve Hataları Dönen mesaj bilgisi daha iyi anlamak için bölüm” [Durum Mesajları \(Report status\)](#)” kısmına bakınız.

7.2.1.1.2 Kırmızı Ledin Kontrol Komutları

Gönderilen Mesaj

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	DATA	CRC (opsiyonel)
0xAA	0x01	0x01	0x01 0x00	NONE	0x0F 0x73

HEADER 0xAA Komutun başlangıcını belirler. Sabit bir değer alır ve tüm mesajların başında bulunur.

DEVICE_ADDRESS 0x01 Cihazın adresini içerir.

COMMAND_TYPE 0x01 Komutun ana türünü belirleyen anahtar koddur, Ledlerin Led Aç/Kapat komutudur.

SUB_TYPE 0x01 led numarasıdır (kırmızı led), 0x00 ise led kapat komudur. Eğer ilgili ledi açmak istenirse 5. Byte hex verisine 0x001 gönderilmez. (Bu durumda ilgili CRC değerinin değiştiğini unutmayın.)

CRC (Opsiyonel) CRC dönen hex değeri 0x0F 0x73.

Gelen mesaj aşağıdadır,

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xBB	0x01	0x00	0x00	0x5C	0x3E

Gönderilen led komuduna göre dönen mesaj başarılı bilgisi içermektedir. Bu mesaj bilgisi “STATUS_CODE_HIGH Ve STATUS_CODE_LOW “anlaşılmaktadır. Genel Sistem Durumu ve Hataları Dönen mesaj bilgisi daha iyi anlamak için bölüm” [Durum Mesajları \(Report status\)](#)” kısmına bakınız.

7.2.2 Cihazdaki Ledlerin LED Toggle Komutları

Bu komut yapısında cihazın üzerindeki led renklerine göre ledlerin LED toggle işlemi yapılmaktadır. Tek renk toggle işlemidir.

Gönderilen mesajın 3. Byte numaralı Hex bilgisi toggle işlemi yapılacağı bilgisini verir. Toggle işlemi için 0x02’dir.

Gönderilen mesajın 4. Byte numaralı Hex bilgisi led numarası bilgisini verir, 5 byte numaralı hex bilgisi ise LED toggle komutu verir her zaman 0x00 dir.

4 numaralı byte bilgisi 0x01 veya 0x03 olabilir, bunun haricinde cihaz hata bilgisi olarak ERR_LED_NUMBER_INVALID bilgisi, bu hata bilgisi için lütfen “[6.4.3 LED ve Göstergeler ile İlgili Hatalar](#)” bakınız.

Hex bilgisi 0x01 numaralı ledin rengi kırmızıdır, hex bilgisi 0x03 numaralı ledin rengi versiyon bilgisine göre MAVİ veya SARI’dır.

7.2.2.1 Mavi veya Sarı Ledin Toggle Komutları

Gönderilen Mesaj

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	DATA	CRC (opsiyonel)
0xAA	0x01	0x02	0x03 0x00	NONE	0x30 0x41

HEADER **0xAA** Komutun başlangıcını belirler. Sabit bir değer alır ve tüm mesajların başında bulunur.

DEVICE_ADDRESS **0x01** Cihazın adresini içerir.

COMMAND_TYPE **0x02** Komutun ana türünü belirleyen anahtar koddur, Ledlerin toggle komutudur.

SUB_TYPE 0x03 led numarasıdır (mavi led), 0x01 ise led aç komudur. Sonraki hex değeri ise 0x00'dir.

Gelen mesaj aşağıdadır,

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xBB	0x01	0x00	0x00	0x5C	0x3E

Gönderilen led komuduna göre dönen mesaj başarılı bilgisi içermektedir. Bu mesaj bilgisi "STATUS_CODE_HIGH

Ve STATUS_CODE_LOW "anlaşılmaktadır. Genel Sistem Durumu ve Hataları Dönen mesaj bilgisi daha iyi anlamak için bölüm" [Durum Mesajları \(Report status\)](#) kısmına bakınız.

7.2.2.2 Kırmızı Ledin Toggle Komutları

Gönderilen Mesaj

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	DATA	CRC (opsiyonel)
0xAA	0x01	0x02	0x01 0x00	NONE	0x56 0x23

HEADER **0xAA** Komutun başlangıcını belirler. Sabit bir değer alır ve tüm mesajların başında bulunur.

DEVICE_ADDRESS **0x01** Cihazın adresini içerir. COMMAND_TYPE **0x02** Komutun ana türünü belirleyen anahtar koddur, Ledlerin toggle komutudur. SUB_TYPE 0x01 led numarasıdır (kırmızı led), 0x00 ise led toggle komudur.

Gelen mesaj aşağıdadır,

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xBB	0x01	0x00	0x00	0x5C	0x3E

Gönderilen led komuduna göre dönen mesaj başarılı bilgisi içermektedir. Bu mesaj bilgisi "STATUS_CODE_HIGH

Ve STATUS_CODE_LOW "anlaşılmaktadır. Genel Sistem Durumu ve Hataları Dönen mesaj bilgisi daha iyi anlamak için bölüm" [Durum Mesajları \(Report status\)](#) kısmına bakınız.

7.2.3 Cihazdaki Ledlerin Çif LED Toggle Komutları

Bu komut yapısında cihazın üzerindeki led renklerine sıra ile toogle yapmaktadır.

Üsteki toogle işleminden farklı olarak burada iki led toogle yapılmaktadır.

Gönderilen mesajın 3. Byte numaralı Hex bilgisi 2 led toogle işlemi yapılacağı bilgisini verir. Toggle işlemi için **0x03**'tür.

Gönderilen mesajın 4. Byte numaralı Hex bilgisi 2li toogle işlemini verir, 0x01 hex bilgisi 2li toggle başla 0x00 ise 2li toggle durdur bilgisini verir.

7.2.3.1 Cihazdaki Ledlerin Çif LED Toggle Başlatma

Gönderilen Mesaj

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	DATA	CRC (opsiyonel)
0xAA	0x01	0x03	0x01 0x00	NONE	0x61 0x13

HEADER **0xAA** Komutun başlangıcını belirler. Sabit bir değer alır ve tüm mesajların başında bulunur.

DEVICE_ADDRESS **0x01** Cihazın adresini içerir.

COMMAND_TYPE **0x03** Komutun ana türünü belirleyen anahtar koddur, Ledlerin ikili toggle komutudur.

SUB_TYPE 0x01 0x00 toogle başlatma komutudur. Sonraki 2 byte'lık değer ise CRC'dir.

Gelen mesaj aşağıdadır,

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xBB	0x01	0x00	0x00	0x5C	0x3E

Gönderilen led komuduna göre dönen mesaj başarılı bilgisi içermektedir. Bu mesaj bilgisi "**STATUS_CODE_HIGH**

Ve **STATUS_CODE_LOW** "anlaşılmaktadır. Genel Sistem Durumu ve Hataları Dönen mesaj bilgisi daha iyi anlamak için bölüm "[Durum Mesajları \(Report status\)](#)" kısmına bakınız.

7.2.3.2 Cihazdaki Ledlerin Çif LED Toggle Durdurma

Gönderilen Mesaj

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	DATA	CRC (opsiyonel)
0xAA	0x01	0x03	0x00 0x00	NONE	0x52 0x22

HEADER **0xAA** Komutun başlangıcını belirler. Sabit bir değer alır ve tüm mesajların başında bulunur.

DEVICE_ADDRESS **0x01** Cihazın adresini içerir.

COMMAND_TYPE **0x03** Komutun ana türünü belirleyen anahtar koddur, Ledlerin ikili toggle komutudur.

SUB_TYPE 0x00 0x00 toggle durdurma komutudur. Sonraki 2 byte'lık değer ise CRC'dir.

Gelen mesaj aşağıdadır,

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xBB	0x01	0x00	0x00	0x5C	0x3E

Gönderilen led komuduna göre dönen mesaj başarılı bilgisi içermektedir. Bu mesaj bilgisi "STATUS_CODE_HIGH

Ve STATUS_CODE_LOW "anlaşılmaktadır. Genel Sistem Durumu ve Hataları Dönen mesaj bilgisi daha iyi anlamak için bölüm "Durum Mesajları (Report status)" kısmına bakınız.

7.2.4 Cihazdaki Tüm Ledleri Kapatma Komutları

Bu komut yapısında cihazın üzerindeki tüm ledleri kapatmak içindir.

Gönderilen mesajın 3. Byte numaralı Hex bilgisi toggle işlemi yapılacağı bilgisini verir. Toggle işlemi için **0x04'dir**.

Gönderilen mesajın 4. Byte numaralı Hex bilgisi ve mesajın 5. Byte numaralı Hex bilgisi 0x00'dir.

Gönderilen Mesaj

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	DATA	CRC (opsiyonel)
0xAA	0x01	0x04	0x00 0x00	NONE	0xD7 0xB2

HEADER **0xAA** Komutun başlangıcını belirler. Sabit bir değer alır ve tüm mesajların başında bulunur.

DEVICE_ADDRESS **0x01** Cihazın adresini içerir.

COMMAND_TYPE **0x04** Komutun ana türünü belirleyen anahtar koddur, Tüm ledlerin kapatma komutudur.

Gelen mesaj aşağıdadır,

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xBB	0x01	0x00	0x00	0x5C	0x3E

Gönderilen led kapat komuduna göre dönen mesaj başarılı bilgisi içermektedir. Bu mesaj bilgisi "STATUS_CODE_HIGH

Ve STATUS_CODE_LOW "anlaşılmaktadır. Genel Sistem Durumu ve Hataları Dönen mesaj bilgisi daha iyi anlamak için bölüm "Durum Mesajları (Report status)" kısmına bakınız.

7.2.5 Cihazdaki Buzzer Aç/Kapatma Komutları

Bu komut yapısında cihazın üzerindeki buzzer açma ve kapatma işlemi yapılmaktadır.

Gönderilen mesajın 3. Byte Hex bilgisi buzzer aç kapat komut bilgisini verir buzzer için **0x05**'tir. 4 numaralı Hex bilgisi buzzer durum bilgisini verir. Durum bilgisi için buzzer aktif komutu 0x01'dir pasif durum ise 0x00'dir.

Cihazdaki buzzer Kontrol Komutları aşağıdadır,

7.2.5.1 Buzzer Aç/Kapat Komutu

Gönderilen Mesaj

Command Sent						
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	DATA	CRC (opsiyonel)	State
0xAA	0x01	0x05	0x01 0x00	NONE	0xD3 0xB3	Buzzer OPEN
0xAA	0x01	0x05	0x00 0x00	NONE	0xE0 0x82	Buzzer CLOSE

HEADER **0xAA** Komutun başlangıcını belirler. Sabit bir değer alır ve tüm mesajların başında bulunur.

DEVICE_ADDRESS **0x01** Cihazın adresini içerir.

COMMAND_TYPE **0x05** Komutun ana türünü belirleyen anahtar koddur, Buzzer Aç/Kapat komutudur.

SUB_TYPE 0x01 0x01'dir Buzzer Aç Komutudur.

SUB_TYPE 0x00 0x00'dir Buzzer Kapat Komutudur.

CRC (Opsiyonel) CRC dönen hex değeri **Buzzer Aç için** 0xD3 0xB3 'dir. **Buzzer Kapat** için 0xE0 0x82'dir.

Gelen mesaj aşağıdadır,

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xBB	0x01	0x00	0x00	0x5C	0x3E

Gönderilen buzzer komuduna göre dönen mesaj başarılı bilgisi içermektedir. Bu mesaj bilgisi "**STATUS_CODE_HIGH**

Ve **STATUS_CODE_LOW** "anlaşılmaktadır. Genel Sistem Durumu ve Hataları Dönen mesaj bilgisi daha iyi anlamak için bölüm" [Durum Mesajları \(Report status\)](#)" kısmına bakınız.

7.2.6 Cihazdaki Süreli Buzzer Aç/Kapatma Komutları

Bu komut yapısında cihazın üzerindeki buzzer süreli olarak açma ve kapatma işlemi yapılmaktadır. 100 ms aralıklarla buzzer programlanabilmektedir.

Max buzzer çalma süresi 25000 ms'dir. Komut sonrası cihazdan dönen "SUCCESS" mesajı diğer komutlardan farklı olarak iki tanedir. İlk olarak buzzer açılması için "SUCCESS" mesajı döner sonra istenilen ms cinsinden buzzer aktif olur sonra buzzer kapatılmas durumunda ise tekrardan "SUCCESS" mesajı gelir.

Gönderilen mesajın 3. Byte Hex bilgisi buzzer aç kapat komut bilgisini verir buzzer için 0x06'tir. 4 numaralı Hex bilgisi her zaman 0x00 dır. 5 numaralı Hex bilgisi Durum bilgisi için buzzer aktif süresi verir. Buradaki hex değerinin desimal karşılığı ile 100 ms çarpılır ve çarpım sonucu kadar buzzer aktif olur ve sonra buzzer kapanır.

Cihazdaki buzzer Kontrol Komutları aşağıdadır,

7.2.6.1 Buzzer Süreli Aç/Kapat Komutu

Gönderilen Mesaj

Command Sent						
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	DATA	CRC (opsiyonel)	State
0xAA	0x01	0x06	0x00 0x05	NONE	0xE9 0x77	Buzzer 500ms
0xAA	0x01	0x06	0x00 0x0A	NONE	0x18 0x98	Buzzer 1000ms
0xAA	0x01	0x06	0x00 0x14	NONE	0xEB 0x67	Buzzer 2000ms
0xAA	0x01	0x06	0x00 0xFA	NONE	0xA7 0x22	Buzzer 25000ms

HEADER 0xAA Komutun başlangıcını belirler. Sabit bir değer alır ve tüm mesajların başında bulunur.

DEVICE_ADDRESS 0x01 Cihazın adresini içerir.

COMMAND_TYPE 0x06 Komutun ana türünü belirleyen anahtar koddur, Buzzer Aç/Kapat komutudur.

SUB_TYPE 0x00 0xXX'dir Süreli Buzzer Aç Komutudur. 0Xxx hex karşılığı ile 100ms çarpımıdır.

Buzzer Süreli Aç/Kapat Komutu Örneği

Hex Value	Decimal Value	Active Buzzer Time
0x05	5 X 100ms	500 ms
0x0A	10 X 100ms	1000ms
0x14	20 X 100ms	2000 ms
0xFF	250 X 100ms	25000 ms

Tablo 34 Buzzer Süreli Aç/Kapat Saniye Hesaplama Örneği

Gelen mesaj aşağıdadır,

Bu mesaj doğru süreli buzzer komutu gönderildiği zaman iki defa gelir ilk olarak Buzzer aktif olduğunda ikinci olarak ise buzzer pasif olduğunda gelir.

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xBB	0x01	0x00	0x00	0x5C	0x3E

Gönderilen buzzer komuduna göre dönen mesaj başarılı bilgisi içermektedir. Bu mesaj bilgisi "STATUS_CODE_HIGH

Ve STATUS_CODE_LOW "anlaşılmaktadır. Genel Sistem Durumu ve Hataları Dönen mesaj bilgisi daha iyi anlamak için bölüm " [Durum Mesajları \(Report status\)](#) " kısmına bakınız.

7.2.7 Led ve Programlabilir Buzzer Tüm Komutları

Bu komutlar Cihaz adresi **0x01** olan CRC hesaplama modu **CRC-16 CCITT** olan cihazlar için verilmiştir.

Komut Adı	Veri Paketi	Açıklama
LED1ON	0xAA 0x01 0x01 0x01 0x01 0x1F 0x52	LED1'i açar.
LED1OFF	0xAA 0x01 0x01 0x01 0x00 0x0F 0x73	LED1'i kapatır.
LED2ON	0xAA 0x01 0x01 0x03 0x01 0x79 0x30	LED2'yi açar.
LED2OFF	0xAA 0x01 0x01 0x03 0x00 0x69 0x11	LED2'yi kapatır.
L.1.TOG	0xAA 0x01 0x02 0x01 0x00 0x56 0x23	LED1'i yanıp sönme (toggle) moduna alır.
L.2.TOG	0xAA 0x01 0x02 0x03 0x00 0x30 0x41	LED2'yi yanıp sönme (toggle) moduna alır.
2LED_TOGGLE_ON	0xAA 0x01 0x03 0x01 0x00 0x61 0x13	Polis ışık efekti başlatır (LED'ler sırayla yanar).
2LED_TOGGLE_OFF	0xAA 0x01 0x03 0x00 0x00 0x52 0x22	Polis ışık efektini kapatır.
LED_ALL_OFF	0xAA 0x01 0x04 0x00 0x00 0xD7 0xB2	Sistemdeki tüm LED'leri kapatır.
BZR_ON	0xAA 0x01 0x05 0x01 0x00 0xD3 0xB3	Buzzer'ı açar (ses çıkarmaya başlar).
BZR_OFF	0xAA 0x01 0x05 0x00 0x00 0xE0 0x82	Buzzer'ı kapatır.
B_1000	0xAA 0x01 0x06 0x00 0x0A 0x18 0x98	Buzzer'ı 1000 ms (1 saniye) çaldırır.
B_500	0xAA 0x01 0x06 0x00 0x05 0xE9 0x77	Buzzer'ı 500 ms (yarım saniye) çaldırır.
B_2000	0xAA 0x01 0x06 0x00 0x14 0xEB 0x67	Buzzer'ı 2000 ms (2 saniye) çaldırır.
B_25000	0xAA 0x01 0x06 0x00 0xFF 0xA7 0x22	Buzzer'ı 25000 ms (25 saniye) çaldırır.

Tablo 35 LED ve Buzzer Kontrol Komutları Tablosu

7.3 RFID Kart İşlem Komutları

RFID kart işlem komutları, kart tanıma, okuma ve yazma işlemlerini gerçekleştirmek amacıyla kullanılan temel komut setini içerir. Bu komutlar, RFID sistemleri ile güvenli ve verimli bir şekilde iletişim kurulmasını sağlar. Aşağıda listelenen komutlar, kartın UID (Unique Identifier) numarasını, SAK (Select Acknowledge) ve ATQA (Answer To Request) değerlerini okumaktan, belirli bloklardan veri okuma ve yazma işlemlerine kadar çeşitli işlevleri kapsamaktadır. Ayrıca, sistemde tanımlı anahtarların görüntülenmesi gibi güvenlik odaklı işlemler de bu komut seti içinde yer almaktadır.

RFID kart işlem komutları, kartların kimlik doğrulama süreçlerinde kullanılmasını sağlarken, aynı zamanda güvenli veri yönetimi için de önemli bir rol oynar. Aşağıda, her bir komutun işlevi ayrıntılı olarak açıklanacaktır.

7.3.1 RFID Kart Tanıma ve Okuma/Yazma Komutları

RFID kart okuma işlemleri, cihazın RFID kartları tanıyabilmesi ve gerekli verileri okuyabilmesi için kullanılan temel komutlardan oluşur. Bu komutlar, kartın UID (Unique Identifier), SAK (Select Acknowledge) ve ATQA (Answer To Request) gibi bilgilerini elde etmeye yöneliktir. Ayrıca, belirli bloklardan veri okuma ve yazma işlemleri de desteklenmektedir.

Komutlar, **HEADER**, **DEVICE_ADDRESS**, **COMMAND_TYPE**, **SUB_COMMAND_TYPE**, **DATA** (yalnızca okuma veya yazma işlemlerinde) ve **CRC** (opsiyonel) bileşenlerinden oluşan dinamik uzunlukta bir yapıdadır. **DATA** alanı, yalnızca RFID kart blok okuma ve yazma komutlarında kullanılır. Veri bütünlüğünü sağlamak için **CRC** alanı isteğe bağlı olarak mesajın sonuna eklenebilir.

Bölüm	Açıklama
HEADER	Komutun başlangıcını belirler. Sabit bir değerdir ve tüm mesajların başında bulunur.
DEVICE_ADDRESS	Hangi cihaza komutun gönderileceğini belirler. Cihazın adresini içerir.(0-255)
COMMAND_TYPE	RFID kart işlemleri için kullanılan ana komut türünü belirler.
SUB_COMMAND_TYPE	RFID komutlarında, UID, SAK veya ATQA okuma gibi farklı alt komutları ifade eder.
DATA (Varsa)	Belirli bloklardan veri okuma ve yazma işlemleri için kullanılır. Okuma işlemlerinde DATA alanı genellikle boş bırakılır, ancak yazma işlemlerinde gerekli veriyi içerir.
CRC (Opsiyonel)	Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır. Eğer aktifse, mesajın sonuna eklenir.

Bu yapı, RFIDAX cihazının RFID kart okuma ve yazma işlemlerini güvenilir ve esnek bir şekilde gerçekleştirmesini sağlar.

Burada dönen cevaplar veri paketleri şeklinde gelir.

Lütfen veri paketleri bölümünü dikkatlice bakınız. ([6.7.2 Veri Paketleri \(send_data_packet\)](#))

HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xAA	0x00-0xFF 1 Byte	Değişken 1 Byte	Değişken Uzunluk	Değişken 1 Byte	Değişken 1 Byte

Not:Bu bölümü incelemenden önce Veri paket yapısı ([6.7.2 Veri Paketleri \(send_data_packet\)](#))ve Mesaj Type([6.7.2.1 Message Type Bilgisi](#)) yapısına bakmanız komut yapısını daha iyi anlamınıza yardımcı olacaktır.

7.3.2 RFID Kart Tanıma Komutları

RFID kartların tanınması ve içerdikleri bilgilerin okunması için belirli alt komutlar kullanılır. Bu komutlar, RFID kartın benzersiz kimlik bilgilerini (UID), güvenlik ile ilgili değerleri (SAK ve ATQA) ve genel kart bilgilerini okumayı sağlar. RFID kart işlemleri **COMMAND_TYPE = 0x07** komutuyla başlar. **SUB_COMMAND_TYPE** alanı ise hangi bilginin okunacağını belirler. Okuma işlemlerinde, UID, SAK ve ATQA değerleri alınabilir veya tüm kart bilgileri tek bir komutla çekilebilir.

Bu komut yapısında **DATA** alanı yalnızca belirli bloklardan veri okuma ve yazma işlemleri için kullanılır. **UID, SAK ve ATQA okuma komutlarında DATA alanı boş bırakılır.** Veri bütünlüğünü sağlamak için **CRC** alanı isteğe bağlı olarak mesajın sonuna eklenebilir.

7.3.2.1 RFID Kart UID Okuma Komutu

Bu komut, RFID kartın benzersiz kimlik numarasını (**UID**) okumak için kullanılır.

Gönderilen Mesaj Formatı

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	DATA	CRC (Opsiyonel)
0xAA	0x01	0x07	0x01	0xFF	0xA3 0x23

Alan Açıklamaları

HEADER (0xAA): Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x07): RFID kart işlemlerini belirler.

SUB_COMMAND_TYPE (0x01): UID okuma işlemini ifade eder.

DATA (0xFF): Bu komutta kullanılmaz, ancak sabit bir dolgu değeri olarak gönderilir.

CRC (0xA3 0x23) (Opsiyonel): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen mesaj aşağıdadır,

HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (Kart ID)	CRC_HIGH (Opsiyonel)	CRC_LOW (Opsiyonel)
0xAA	0x01	0x07	0x88 0x04 0xDE 0x18	0xC4	0xD1

HEADER (0xAA): Mesajın başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Yanıtı gönderen cihazın adresini gösterir.

MESSAGE_TYPE (0x07): RFID kart işlem yanıtı olduğunu belirtir.

DATA (Kart ID): 0x88 0x04 0xDE 0x18 (Kart kimlik bilgisi).

CRC_HIGH (0xC4) & CRC_LOW (0xD1): Opsiyonel hata kontrol değerleri.

Bu yanıt, RFID kartın UID bilgisinin başarıyla okunduğunu gösterir.

Bu başarılı bir RFID tag okuma işlemidir. Mesaj type bilgisi anlamlandırabilmek için lütfen "[6.7.2.1.1 RFID Kart Okuma İşlemleri Mesajları ve Message Type Bilgisi](#)" bakınız.



Advanced RFID Card Reader Writer

7.3.2.2 RFID Kart SAK Değeri Okuma Komutu

Bu komut, RFID kartın **SAK (Select Acknowledge)** değerini okumak için kullanılır. SAK değeri, kartın tipini ve desteklediği protokolleri belirlemek için kullanılan bir kimlik doğrulama bilgisidir.

Gönderilen Mesaj Formatı

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	DATA	CRC (Opsiyonel)
0xAA	0x01	0x07	0x02	0xFF	0xF6 0x70

Alan Açıklamaları

HEADER (0xAA): Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x07): RFID kart işlemlerini belirler.

SUB_COMMAND_TYPE (0x02): SAK okuma işlemini ifade eder.

DATA (0xFF): Bu komutta kullanılmaz, ancak sabit bir dolgu değeri olarak gönderilir.

CRC (0xF6 0x70) (Opsiyonel): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (Kart SAK Değeri)	CRC_HIGH (Opsiyonel)	CRC_LOW (Opsiyonel)
0xAA	0x01	0x09	0x08	0x0A	0xBD

Alan Açıklamaları

HEADER (0xAA): Mesajın başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Yanıtı gönderen cihazın adresini gösterir.

MESSAGE_TYPE (0x09): RFID kart işlem yanıtı olduğunu belirtir.

DATA (Kart SAK Değeri): 0x08 (Kartın SAK değeri).

CRC_HIGH (0x0A) & CRC_LOW (0xBD): Opsiyonel hata kontrol değerleri.

Bu yanıt, RFID kartın **SAK bilgisinin başarıyla okunduğunu** gösterir.

Bu başarılı bir RFID tag SAK Değeri Okuma işlemidir. **MESSAGE_TYPE** bilgisini anlamlandırabilmek için lütfen

[“6.7.2.1.1 RFID Kart Okuma İşlemleri Mesajları ve Message Type Bilgisi”](#) bölümüne bakınız.



Advanced RFID Card Reader Writer

7.3.2.3 RFID Kart ATQA Değeri Okuma Komutu

Bu komut, RFID kartın **ATQA (Answer To Request)** değerini okumak için kullanılır. ATQA değeri, kartın türünü ve haberleşme özelliklerini belirleyen önemli bir kimlik bilgisidir.

Gönderilen Mesaj Formatı

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	DATA	CRC (Opsiyonel)
0xAA	0x01	0x07	0x03	0xFF	0xC5 0x41

Alan Açıklamaları

HEADER (0xAA): Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x07): RFID kart işlemlerini belirler.

SUB_COMMAND_TYPE (0x03): ATQA okuma işlemini ifade eder.

DATA (0xFF): Bu komutta kullanılmaz, ancak sabit bir dolgu değeri olarak gönderilir.

CRC (0xC5 0x41) (Opsiyonel): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (Kart ATQA Değeri)	CRC_HIGH (Opsiyonel)	CRC_LOW (Opsiyonel)
0xAA	0x01	0x08	0x00 0x04	0xE2	0x57

Alan Açıklamaları

HEADER (0xAA): Mesajın başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Yanıtı gönderen cihazın adresini gösterir.

MESSAGE_TYPE (0x08): RFID kart işlem yanıtı olduğunu belirtir.

DATA (Kart ATQA Değeri): 0x00 0x04 (Kartın ATQA değeri).

CRC_HIGH (0xE2) & CRC_LOW (0x57): Opsiyonel hata kontrol değerleri.

Bu yanıt, RFID kartın **ATQA bilgisinin başarıyla okunduğunu** gösterir.

Bu başarılı bir RFID tag ATQA Değeri Okuma işlemidir. **MESSAGE_TYPE** bilgisini anlamlandırabilmek için lütfen

[“6.7.2.1.1 RFID Kart Okuma İşlemleri Mesajları ve Message Type Bilgisi”](#) bölümüne bakınız.



Advanced RFID Card Reader Writer

7.3.2.4 Kart bilgilerini okuma (UID + SAK + ATQA) Komutu

Bu komut, RFID kartın tüm kimlik bilgilerini tek seferde okumak için kullanılır. **UID (Unique Identifier)**, **SAK (Select Acknowledge)** ve **ATQA (Answer To Request)** bilgilerini içeren kapsamlı bir yanıt döndürülür.

Gönderilen Mesaj Formatı

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	DATA	CRC (Opsiyonel)
0xAA	0x01	0x07	0x04	0xFF	0x5C 0xD6

Alan Açıklamaları

HEADER (0xAA): Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x07): RFID kart işlemlerini belirler.

SUB_COMMAND_TYPE (0x04): Kart bilgilerini okuma işlemini ifade eder.

DATA (0xFF): Bu komutta kullanılmaz, ancak sabit bir dolgu değeri olarak gönderilir.

CRC (0x5C 0xD6) (Opsiyonel): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (Kart UID + SAK + ATQA)	CRC_HIGH (Opsiyonel)	CRC_LOW (Opsiyonel)
0xAA	0x01	0x0A	0x66 0xA7 0x7B 0xDA 0x08 0x00 0x04	0x31	0x3C

Alan Açıklamaları

HEADER (0xAA): Mesajın başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Yanıtı gönderen cihazın adresini gösterir.

MESSAGE_TYPE (0x0A): RFID kart işlem yanıtı olduğunu belirtir.

DATA (Kart UID + SAK + ATQA):

0x66 0xA7 0x7B 0xDA → Kart UID (Benzersiz Kimlik Numarası)

0x08 → Kart SAK Değeri

0x00 0x04 → Kart ATQA Değeri

CRC_HIGH (0x31) & CRC_LOW (0x3C): Opsiyonel hata kontrol değerleri.

Bu yanıt, RFID kartın **UID, SAK ve ATQA bilgilerini başarıyla okunduğunu** gösterir.

Bu başarılı bir RFID tag okuma işlemidir. **MESSAGE_TYPE** bilgisini anlamlandırabilmek için lütfen "[6.7.2.1.1 RFID Kart Okuma İşlemleri Mesajları ve Message Type Bilgisi](#)" bölümüne bakınız.



Advanced RFID Card Reader Writer

7.3.2.5 RFID Kart Anahtarlarını Gösterme Komutu

Bu komut, RFID kartın güvenlik anahtarlarını (KeyA, KeyB, Optional KeyA, Optional KeyB) görüntülemek için kullanılır. RFID sistemlerinde, kart bloklarına erişim belirli anahtarlarla yapılır. Bu komut, kayıtlı anahtar bilgilerini okumak ve doğrulamak için kullanılır.

Eğer SUB_COMMAND_TYPE değeri:

- 0x05 0x01 → KeyA görüntülenir.
- 0x05 0x02 → KeyB görüntülenir.
- 0x05 0x03 → Optional KeyA görüntülenir.
- 0x05 0x04 → Optional KeyB görüntülenir.
- 0x05 0xFF → Tüm anahtarlar gösterilir.

KeyA ve KeyB, RFID sistemlerinde sabit anahtarlardır ve değiştirilemez.

Eğer özel bir anahtar kullanılması gerekiyorsa, Optional KeyA ve Optional KeyB kullanıcı tarafından kaydedilebilir, değiştirilebilir ve okunabilir.

Gönderilen Mesaj Formatı

Command Sent						
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	DATA (Not Used)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x07	0x05 XX	Not Used	XX	XX

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x07) → RFID kart işlemlerini belirler.

SUB_COMMAND_TYPE → Hangi anahtarın okunacağını belirler:

- 0x05 0x01 → KeyA
- 0x05 0x02 → KeyB
- 0x05 0x03 → Optional KeyA
- 0x05 0x04 → Optional KeyB
- 0x05 0xFF → Tüm anahtarları göster

DATA → Bu komutta kullanılmaz.

CRC (XX XX) (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

7.3.2.5.1 Sabit Anahtarlar (KeyA ve KeyB)

RFID sistemlerinde, kart bloklarına erişimi kontrol etmek için **KeyA ve KeyB** kullanılır. Bu anahtarlar **sabit olup değiştirilemez ve sistemde varsayılan olarak tanımlanmıştır**.

Anahtar Türü	Varsayılan Değer
KeyA	0xFF 0xFF 0xFF 0xFF 0xFF 0xFF
KeyB	0x00 0x00 0x00 0x00 0x00 0x00

KeyA ve KeyB, üretici tarafından belirlenen sabit anahtarlardır ve değiştirilemez. Genellikle varsayılan erişim anahtarları olarak kullanılırlar.

KeyA Okuma

KeyA'yı okumak için aşağıdaki komut yapısını inceleyiniz.

Giden Mesaj (KeyA Okuma - Sabit Anahtar)

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	CRC_HIGH	CRC_LOW
0xAA	0x01	0x07	0x05 0x01	0x61	0x36

Gelen Mesaj (KeyA Yanıtı - Sabit Anahtar)

Response					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (KeyA - Sabit Anahtar)	CRC_HIGH	CRC_LOW
0xAA	0x01	0x0B	0xFF 0xFF 0xFF 0xFF 0xFF 0xFF	0x74	0x87

KeyB Okuma

KeyB'yi okumak için aşağıdaki komut yapısını inceleyiniz.

Giden Mesaj (KeyB Okuma - Sabit Anahtar)

Command Sent

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	CRC_HIGH	CRC_LOW
0xAA	0x01	0x07	0x05 0x02	0x51	0x55

Gelen Mesaj (KeyB Yanıtı - Sabit Anahtar)

Response					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (KeyB - Sabit Anahtar)	CRC_HIGH	CRC_LOW
0xAA	0x01	0x0C	0x00 0x00 0x00 0x00 0x00 0x00	0xFA	0x1C

7.3.2.5.2 Kullanıcı Tarafından Tanımlanabilen Anahtarlar (Optional KeyA ve Optional KeyB)

Özel güvenlik gereksinimleri için kullanıcı tarafından tanımlanabilen anahtarlar mevcuttur.

Bu anahtarlar **kullanıcı tarafından kaydedilebilir, değiştirilebilir ve okunabilir.**

Anahtar Türü	Örnek Kullanıcı Kaydedilen Değer
Optional KeyA	0x1A 0x2A 0x3A 0x4A 0x5A 0x6A
Optional KeyB	0xA1 0xB2 0x99 0xD1 0xE1 0xF1

Optional KeyA ve Optional KeyB, kullanıcı tarafından kaydedilebilir ve değiştirilebilir.

Eğer özel bir KeyA veya KeyB kullanılacaksa, opsiyonel anahtarlar tercih edilmelidir.

Bu anahtarların değiştirilme işlemleri, Bellek Yönetim Komutları bölümünde anlatılmıştır. Lütfen bakınız ([7.5.2 OptionalKeyA ve OptionalKeyB İşlemleri](#))

Optional KeyA Okuma

Optional KeyA'yı okumak için aşağıdaki komut yapısını inceleyiniz.

Giden Mesaj (Optional KeyA Okuma - Kullanıcı Kaydedilmiş Anahtar)

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	CRC_HIGH	CRC_LOW
0xAA	0x01	0x07	0x05 0x03	0x41	0x74

Gelen Mesaj (Optional KeyA Yanıtı - Kullanıcı Kaydedilmiş Anahtar)

Response					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (Optional KeyA- Kullanıcı Kaydedilen)	CRC_HIGH	CRC_LOW
0xAA	0x01	0x0E	0x1A 0x2A 0x3A 0x4A 0x5A 0x6A	0xFA	0xD5

Optional KeyB Okuma

Optional KeyB'yi okumak için aşağıdaki komut yapısını inceleyiniz.

Giden Mesaj (Optional KeyB Okuma - Kullanıcı Kaydedilmiş Anahtar)

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	CRC_HIGH	CRC_LOW
0xAA	0x01	0x07	0x05 0x04	0x31	0x93

Gelen Mesaj (Optional KeyB Yanıtı - Kullanıcı Kaydedilmiş Anahtar)

Response					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (Optional KeyB- Kullanıcı Kaydedilen)	CRC_HIGH	CRC_LOW
0xAA	0x01	0x0F	0xA1 0xB2 0x99 0xD1 0xE1 0xF1	0x57	0x5A



Advanced RFID Card Reader Writer

7.3.2.5.3 Tüm Anahtarları Okuma (KeyA, KeyB, Optional KeyA ve Optional KeyB)

Tüm anahtarları (KeyA, KeyB, Optional KeyA, Optional KeyB) okumak için aşağıdaki komut yapısını inceleyiniz.

Giden Mesaj (Tüm Anahtarları Okuma- FF Kullanımı)

Command Sent					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	CRC_HIGH	CRC_LOW
0xAA	0x01	0x07	0x05 0xFF	0x6F	0xE7

Gelen Mesaj (Tüm Anahtarları Getiren Yanıt- FF Kullanımı)

Bu komut gönderildiğinde, cihaz aşağıdaki sırayla dört ayrı mesaj döndürür:

Yanıt Tipi	Mesaj
KeyA Yanıtı	0xAA 0x01 0x0B 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0x74 0x87
KeyB Yanıtı	0xAA 0x01 0x0C 0x00 0x00 0x00 0x00 0x00 0x00 0xFA 0x1C
Optional KeyA Yanıtı	0xAA 0x01 0x0E 0x1A 0x2A 0x3A 0x4A 0x5A 0x6A 0xFA 0xD5
Optional KeyB Yanıtı	0xAA 0x01 0x0F 0xA1 0xB2 0x99 0xD1 0xE1 0xF1 0x57 0x5A

Bu anahtarların değiştirilme işlemleri, Bellek Yönetim Komutları bölümünde anlatılmıştır. Lütfen bakınız [\(7.5.2 OptionalKeyA ve OptionalKeyB İşlemleri\)](#)



Advanced RFID Card Reader Writer

7.3.3 RFID Kart İçindeki Blokların İçine Veri Yazma ve Bloktan Veri Okuma İşlemi

RFID kartlar, belirli hafıza blokları içeren veri depolama alanlarına sahiptir. Bu bloklara veri yazılabilir, okunabilir ve belirli erişim seviyeleri ile korunabilir.

Bu bölümde, RFID kartlarındaki bloklara veri yazma ve bloklardan veri okuma işlemleri detaylandırılacaktır. RFID blok okuma ve yazma işlemleri belirli kimlik doğrulama gereksinimlerine sahiptir ve KeyA veya KeyB kullanılarak erişim sağlanır.

Her blok, belirli bir güvenlik yapısına sahiptir ve kimlik doğrulama (authentication) gerektirir. Bloklara erişim ve yazma izinleri, kartın güvenlik yapılandırmasına bağlıdır.

7.3.3.1 RFID Kart Bloklarına Veri Yazma Komutu

Bu komut, RFID kartın belirli bloklarına veri yazmak için kullanılır. Yazma işlemi sırasında, kimlik doğrulama (KeyA, KeyB, opsiyonel KeyA, opsiyonel KeyB ile) yapılması zorunludur.

Yazma işlemi sırasında başlangıç ve bitiş blok adresleri belirlenmeli, kartın güvenlik yapısına uygun olup olmadığı kontrol edilmelidir.

Eğer hatalı bir blok seçilirse veya yazılacak veri izin verilen sınırları aşarsa, işlem reddedilir ve hata mesajı döndürülür.

Gönderilen Mesaj Formatı

Command Sent											
HEADER	DEVICE_ ADDRESS	COMMAND_ TYPE	SUB_ COMMAND_ TYPE	KEY _TYPE	AUTH _TYPE	START _BLOCK	END _BLOCK	FORMAT_ TYPE	DATA (Variable Length)	CRC_ HIGH (Optional)	CRC_ LOW (Optional)
0xAA	0x01	0x09	0xFF	0x01 0x02 0x03 or 0x04	0x01 or 0x02	0xFF	0xFF	0x00	User Data	XX	XX

Alan Açıklamaları

HEADER (0xAA): Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x09): RFID kart işlemlerini belirler. Bu komut blok yazma işlemini ifade eder.

SUB_COMMAND_TYPE (0xFF): Bu komutta kullanılmaz.

KEY_TYPE (0x01, 0x02, 0x03, 0x04): Kullanılacak anahtar türünü belirler.

0x01 → KeyA

0x02 → KeyB

0x03 → Opsiyonel_KeyA

0x04 → Opsiyonel_KeyB

AUTH_TYPE (0x01 veya 0x02): Kimlik doğrulama yöntemini belirler.

0x01 → PICC_AUTHENT1A

0x02 → PICC_AUTHENT1B

START_BLOCK (0xFF): Yazılacak ilk blok adresini belirtir.

END_BLOCK (0xFF): Yazılacak son blok adresini belirtir. (Eğer tek bir blok yazmak isterseniz START_BLOCK ve END_BLOCK aynı olmalıdır)

FORMAT_TYPE (0x00 veya 0x01): Verinin formatını belirler.

0x00 → HEX formatı

0x01 → ASCII formatı (Bu format kullanılmamaktadır)

DATA (Variable Length): Kullanıcı tarafından yazılacak veri.

CRC_HIGH (XX) & CRC_LOW (XX) (Opsiyonel): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (opsiyonel)	CRC_LOW (opsiyonel)
0xBB	0x01	0x00	0x00	0x5C	0x3E

Alan Açıklamaları

HEADER (0xAA): Mesajın başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Yanıtı gönderen cihazın adresini gösterir.

STATUS:

0x00 → Başarıyla yazıldı

Hata Kodu → Yazma işlemi başarısız oldu

CRC_HIGH (XX) & CRC_LOW (XX) (Opsiyonel): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gönderilen write işlemine göre dönen mesaj başarılı bilgisi içermektedir. Bu mesaj bilgisi "STATUS_CODE_HIGH Ve STATUS_CODE_LOW" anlaşılmaktadır. Genel Sistem Durumu ve Hataları Dönen mesaj bilgisi daha iyi anlamak için bölüm "6.4.4 RFID Kart Okuma/Yazma Hataları" kısmına bakınız.

Bu Komutun Çalışma Prensipleri

1. Kartın var olup olmadığı kontrol edilir. Eğer kart algılanmazsa, **ERR_CARD_NOT_FOUND** hatası döndürülür.
2. Başlangıç ve bitiş blokları doğrulanır. Eğer başlangıç bloğu, bitiş bloğundan büyükse, **ERR_INVALID_BLOCK_RANGE** hatası döner.
3. Kimlik doğrulama yapılır. KeyA, KeyB, opsiyonel KeyA veya opsiyonel KeyB ile kimlik doğrulaması başarısız olursa, **ERR_AUTHENTICATION_FAILED** hatası döner.
4. Yazılacak veri formatı kontrol edilir:
HEX formatındaysa, verinin doğrudan yazılmasına izin verilir.
5. Veri uzunluğu kontrol edilir. Seçilen bloklara sığmayacak kadar büyük bir veri varsa, **ERR_DATA_LENGTH_EXCEEDED** hatası döndürülür.
6. Yazma işlemi başlatılır.
Seçilen bloklara 16 baytlık veri yazılır.
Blok 0 ve her sektörün kontrol blokları (örneğin 3, 7, 11) atlanır.
7. İşlem tamamlanınca kart durdurulur ve modül yeniden başlatılır.

Olası Hata Kodları

Hata Kodu	Açıklama
ERR_CARD_NOT_FOUND	Kart algılanamadı, işlem yapılamaz.
ERR_INVALID_BLOCK_RANGE	Geçersiz blok aralığı seçildi.
ERR_INVALID_DATA	Yazılacak veri hatalı veya eksik.
ERR_DATA_LENGTH_EXCEEDED	Yazılacak veri çok uzun, seçilen bloklara sığmıyor.
ERR_AUTHENTICATION_FAILED	Kimlik doğrulama başarısız oldu, yetkilendirme yapılamadı.

Özet

Bu komut, RFID kartın belirlenen bloklarına veri yazmak için kullanılır.

KeyA veya KeyB, opsiyonel KeyA veya opsiyonel KeyB ile kimlik doğrulaması gerektirir.

Veri HEX formatında olmalıdır.

Geçersiz blok aralığı, yetkilendirme hatası veya veri boyutu hataları işlem başarısızlığına neden olabilir.

Bu komut, gelişmiş erişim kontrolü ve veri saklama gereksinimleri için kritik öneme sahiptir.



Advanced RFID Card Reader Writer

7.3.3.1.1 RFID Kart Bloklarına Veri Yazma Komutu Örneği

Bu örnek, RFID kartın belirli bir bloğuna veri yazma işlemini göstermektedir.

Kimlik doğrulama (KeyA, KeyB, opsiyonel_KeyA, opsiyonel_KeyB), başlangıç ve bitiş blok adresleri ve veri formatı belirlenerek veri yazma işlemi gerçekleştirilir.

Gönderilen Mesaj Formatı

Command Sent (Command to Write Data to RFID Card Blocks)										
HEADER	DEVICE _ADDRESS	COMMAND _TYPE	KEY _TYPE	AUTH _TYPE	START _BLOCK	END _BLOCK	FORMAT _TYPE	DATA (Variable Length)	CRC _HIGH (Optional)	CRC _LOW (Optional)
0xAA	0x01	0x09	0x01	0x02	0x21	0x21	0x00	0x52 0x46 0x49 0x44 0x41 0x58 0x20 0x43 0x69 0x68 0x61 0x7A 0x6C 0x61 0x72 0x69	0x95	0xD9

DATA (Variable Length) Açıklamaları

HEX Format	ASCII Karşılığı
0x52	R
0x46	F
0x49	I
0x44	D
0x41	A
0x58	X
0x20	(Boşluk)
0x43	C
0x69	i
0x68	h
0x61	a
0x7A	z
0x6C	l
0x61	a
0x72	r
0x69	i

HEX Format: 0x52 0x46 0x49 0x44 0x41 0x58 0x20 0x43 0x69 0x68 0x61 0x7A 0x6C 0x61 0x72 0x69

ASCII Karşılığı: "RFIDAX Cihazları"

Alan Açıklamaları

HEADER (0xAA): Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x09): RFID kart işlemlerini belirler (RFID blok yazma komutu).

KEY_TYPE (0x01): KeyA kullanılarak doğrulama yapılacağını belirtir.

AUTH_TYPE (0x02): PICC_AUTHENT1B kimlik doğrulama yöntemini belirler.

START_BLOCK (0x21): Yazılacak ilk blok adresini belirtir.

END_BLOCK (0x21): Yazılacak son blok adresini belirtir (sadece tek bir bloğa yazılıyor).

FORMAT_TYPE (0x00): Veri formatı HEX olarak seçilmiş.

DATA: "RFIDAX Cihazları" (ASCII karakterleri olarak temsil edilen HEX format)

CRC_HIGH (0x95) & CRC_LOW (0xD9) (Opsiyonel): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.



Advanced RFID Card Reader Writer

Gelen Mesaj Formatı

Response (RFID Kart Blok Yazma Yanıtı)

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xBB	0x01	0x00	0x00	0x5C	0x3E

Alan Açıklamaları

HEADER (0xBB): Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Yanıtı gönderen cihazın adresini gösterir.

STATUS_CODE_HIGH (0x00): İşlem sonucunu belirten ilk bayt.

STATUS_CODE_LOW (0x00): İşlem sonucunu belirten ikinci bayt.

0x00 0x00 → Başarıyla yazıldı

CRC_HIGH (0x5C) & CRC_LOW (0x3E) (Opsiyonel): Veri bütünlüğünü doğrulamak için kullanılan hata kontrol mekanizmasıdır.

Bu Örnekte Gerçekleşen İşlemler

RFID kartın mevcut olup olmadığı kontrol edilmiştir.

KeyA ve AUTH_TYPE = PICC_AUTHENT1B kullanılarak kimlik doğrulama yapılmıştır.

Blok 0x21'e "RFIDAX Cihazlari" metni HEX formatında yazılmıştır.

Cihaz tarafından yazma işlemi başarıyla tamamlanmış ve yanıt olarak 0x00 0x00 (başarı) mesajı döndürülmüştür.

Özet

Bu örnekte RFID kartın 0x21 numaralı bloğuna ASCII formatında "RFIDAX Cihazlari" verisi yazılmıştır.

Cihaz, yazma işlemini başarıyla tamamlamış ve 0x00 0x00 başarı kodunu içeren bir yanıt döndürmüştür.

Bu mesaj formatı, RFID blok yazma işlemlerinin nasıl yapıldığını anlamak için önemli bir referans niteliğindedir.



Advanced RFID Card Reader Writer

7.3.3.1.2 RFID Kart Bloklarına Veri Yazma Komutu Örneği -2

Bu örnek, RFID kartın belirli bir blok aralığına veri yazma işlemini göstermektedir.

Kimlik doğrulama (KeyA, KeyB, opsiyonel_KeyA, opsiyonel_KeyB), başlangıç ve bitiş blok adresleri ve veri formatı belirlenerek veri yazma işlemi gerçekleştirilir.

Gönderilen Mesaj Formatı

Command Sent (RFID Kart Bloklarına Veri Yazma Komutu)

AA 01 09 01 02 04 06 00 4F 6E 75 72 20 4B 61 67 61 6E 20 45 73 6D 65 72 61 79 20 48 61 72 64 77 61 72 65 20 45 6E 67 69 6E 65 65 72 20 20 EC DB

HEADER	DEVICE _ADDRESS	COMMAND _TYPE	KEY_TYPE	AUTH _TYPE	START _BLOCK	END _BLOCK	FORMAT _TYPE	DATA (Variable Length)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x09	0x01	0x02	0x04	0x06	0x00	See Below	0xEC	0xDB

DATA (Variable Length) Açıklamaları

HEX Format:

0x4F 0x6E 0x75 0x72 0x20 0x4B 0x61 0x67 0x61 0x6E 0x20 0x45 0x73 0x6D 0x65
0x72 0x61 0x79 0x20 0x48 0x61 0x72 0x64 0x77 0x61 0x72 0x65 0x20 0x45 0x6E
0x67 0x69 0x6E 0x65 0x65 0x72 0x20 0x20

ASCII Karşılığı: "Onur Kagan Esmeray Hardware Engineer "

Alan Açıklamaları

HEADER (0xAA): Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x09): RFID kart işlemlerini belirler (RFID blok yazma komutu).

KEY_TYPE (0x01): KeyA kullanılarak doğrulama yapılacağını belirtir.

AUTH_TYPE (0x02): PICC_AUTHENT1B kimlik doğrulama yöntemini belirler.

START_BLOCK (0x04): Yazılacak ilk blok adresini belirtir.

END_BLOCK (0x06): Yazılacak son blok adresini belirtir.

FORMAT_TYPE (0x00): Veri formatı HEX olarak seçilmiş.

DATA: "Onur Kagan Esmeray Hardware Engineer " (ASCII karakterleri olarak temsil edilen HEX format)

CRC_HIGH (0xEC) & CRC_LOW (0xDB) (Opsiyonel): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.



Advanced RFID Card Reader Writer

Gelen Mesaj Formatı

Response (RFID Kart Blok Yazma Yanıtı)

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xBB	0x01	0x00	0x00	0x5C	0x54

Alan Açıklamaları

HEADER (0xBB): Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Yanıtı gönderen cihazın adresini gösterir.

STATUS_CODE_HIGH (0x00): İşlem sonucunu belirten ilk bayt.

STATUS_CODE_LOW (0x00): İşlem sonucunu belirten ikinci bayt.

0x00 0x00 → Başarıyla yazıldı

CRC_HIGH (0x5C) & CRC_LOW (0x54) (Opsiyonel): Veri bütünlüğünü doğrulamak için kullanılan hata kontrol mekanizmasıdır.

Bu Örnekte Gerçekleşen İşlemler

RFID kartın mevcut olup olmadığı kontrol edilmiştir.

KeyA ve AUTH_TYPE = PICC_AUTHENT1B kullanılarak kimlik doğrulama yapılmıştır.

Blok 0x04 ile 0x06 arasına "Onur Kagan Esmeray Hardware Engineer " metni HEX formatında yazılmıştır.

Cihaz tarafından yazma işlemi başarıyla tamamlanmış ve yanıt olarak 0x00 0x00 (başarı) mesajı döndürülmüştür.

Özet

Bu örnekte RFID kartın 0x04 - 0x06 blokları arasına ASCII formatında "Onur Kagan Esmeray Hardware Engineer " verisi yazılmıştır.

Cihaz, yazma işlemini başarıyla tamamlamış ve 0x00 0x00 başarı kodunu içeren bir yanıt döndürmüştür.

Bu mesaj formatı, RFID blok yazma işlemlerinin nasıl yapıldığını anlamak için önemli bir referans niteliğindedir.



Advanced RFID Card Reader Writer

7.3.3.1.3 RFID Kart Bloklarına Veri Yazma Komutu Örneği -3

Bu örnek, RFID kartın birden fazla bloğuna uzun veri yazma işlemini göstermektedir.

Kimlik doğrulama (KeyA ,KeyB, opsiyonel_KeyA, opsiyonel_KeyB), başlangıç ve bitiş blok adresleri belirlenerek, HEX formatında uzun metin verisi yazılmaktadır.

Gönderilen Mesaj Formatı

Command Sent (RFID Kart Bloklarına Uzun Veri Yazma Komutu)

```
AA 03 09 01 02 10 1A 00 52 46 49 44 41 58 20 63 69 68 61 7A 6C 61 72 69 20 73
65 6B 74 6F 72 75 6E 20 65 6E 20 69 79 69 20 76 65 20 65 6E 20 74 65 6B 6E 6F
6C 6F 6A 69 6B 20 63 69 68 61 7A 6C 61 72 69 64 69 72 2E 0A 62 69 72 63 6F 6B
20 6D 6F 64 65 6C 69 20 62 75 6C 75 6E 6D 61 6B 74 61 64 69 72 2E 53 69 70 61
72 69 73 20 69 63 69 6E 20 77 77 77 2E 72 66 69 64 61 78 2E 63 6F 6D 20 7A 69
79 61 72 65 74 20 65 64 69 6E 69 7A 2E 4D D1
```

Command Sent

HEADER	DEVICE_ADDRESS	COMMAND_TYPE	KEY_TYPE	AUTH_TYPE	START_BLOCK	END_BLOCK	FORMAT_TYPE	DATA (Variable Length)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x03	0x09	0x01	0x02	0x10	0x1A	0x00	See Below	0x4D	0xD1

DATA (Variable Length) Açıklamaları

HEX Format:

```
0x52 0x46 0x49 0x44 0x41 0x58 0x20 0x63 0x69 0x68 0x61 0x7A 0x6C 0x61 0x72 0x69
0x20 0x73 0x65 0x6B 0x74 0x6F 0x72 0x75 0x6E 0x20 0x65 0x6E 0x20 0x69 0x79 0x69
0x20 0x76 0x65 0x20 0x65 0x6E 0x20 0x74 0x65 0x6B 0x6E 0x6F 0x6C 0x6F 0x6A 0x69
0x6B 0x20 0x63 0x69 0x68 0x61 0x7A 0x6C 0x61 0x72 0x69 0x64 0x69 0x72 0x2E 0x0A
0x62 0x69 0x72 0x63 0x6F 0x6B 0x20 0x6D 0x6F 0x64 0x65 0x6C 0x69 0x20 0x62 0x75
0x6C 0x75 0x6E 0x6D 0x61 0x6B 0x74 0x61 0x64 0x69 0x72 0x2E 0x53 0x69 0x70 0x61
0x72 0x69 0x73 0x20 0x69 0x63 0x69 0x6E 0x20 0x77 0x77 0x77 0x2E 0x72 0x66 0x69
0x64 0x61 0x78 0x2E 0x63 0x6F 0x6D 0x20 0x7A 0x69 0x79 0x61 0x72 0x65 0x74 0x20
0x65 0x64 0x69 0x6E 0x69 0x7A 0x2E
```

ASCII Karşılığı:

"RFIDAX cihazları sektörün en iyi ve en teknolojik cihazlarıdır.

Birçok modeli bulunmaktadır. Sipariş için www.rfidax.com ziyaret ediniz."

Alan Açıklamaları

HEADER (0xAA): Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x03): Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x09): RFID kart işlemlerini belirler (RFID blok yazma komutu).

KEY_TYPE (0x01): KeyA kullanılarak doğrulama yapılacağını belirtir.

AUTH_TYPE (0x02): PICC_AUTHENT1B kimlik doğrulama yöntemini belirler.

START_BLOCK (0x10): Yazılacak ilk blok adresini belirtir.

END_BLOCK (0x1A): Yazılacak son blok adresini belirtir.

FORMAT_TYPE (0x00): Veri formatı HEX olarak seçilmiştir.

DATA: Uzun metin mesajı ASCII formatında kaydedilmiştir.

CRC_HIGH (0x4D) & CRC_LOW (0xD1) (Opsiyonel): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.



Advanced RFID Card Reader Writer

Gelen Mesaj Formatı

Response (RFID Kart Blok Yazma Yanıtı)

Response					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xBB	0x03	0x00	0x00	0x32	0x5E

Alan Açıklamaları

HEADER (0xBB): Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x03): Yanıtı gönderen cihazın adresini gösterir.

STATUS_CODE_HIGH (0x00): İşlem sonucunu belirten ilk bayt.

STATUS_CODE_LOW (0x00): İşlem sonucunu belirten ikinci bayt.

0x00 0x00 → Başarıyla yazıldı

CRC_HIGH (0x32) & CRC_LOW (0x5E) (Opsiyonel): Veri bütünlüğünü doğrulamak için kullanılan hata kontrol mekanizmasıdır.

Özet

Bu örnekte RFID kartın 0x10 - 0x1A blokları arasına ASCII formatında uzun bir metin yazılmıştır.

Cihaz, yazma işlemini başarıyla tamamlamış ve 0x00 0x00 başarı kodunu içeren bir yanıt döndürmüştür.

Bu mesaj formatı, RFID kartlara uzun veri yazma işlemlerini anlamak için önemli bir referans niteliğindedir.



Advanced RFID Card Reader Writer

7.3.3.2 RFID Kart Bloklarına Veri Okuma Komutu

RFID kartlarda belirli bloklara yazılmış verileri okumak için kullanılan komuttur. Okuma işlemi, **kimlik doğrulama (KeyA, KeyB, Opsiyonel KeyA, Opsiyonel KeyB)** gerektirir ve kullanıcının belirttiği **blok adreslerinden** veri çekilerek yanıt olarak geri gönderilir.

Bu komut, **tek bir blok** veya **belirli bir blok aralığını** okumak için kullanılabilir. **Başlangıç bloğu** ve **bitiş bloğu** belirlenerek kart içerisindeki veriler okunur.

Eğer blok aralığı **geçersiz** ise, yani başlangıç bloğu bitiş bloğundan büyükse, hata mesajı döndürülür.

Okuma işlemi tamamlandıktan sonra, sistem **okunan blok verilerini 16 baytlık paketler** halinde döndürür.

Gönderilen Mesaj Formatı

Command Sent (RFID Kart Bloklarına Veri Okuma Komutu)

Command Sent (Command to Read Data to RFID Card Blocks)								
HEADER	DEVICE _ADDRESS	COMMAND _TYPE	KEY _TYPE	AUTH _TYPE	START _BLOCK	END _BLOCK	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x08	0x01 - 0x04	0x01 - 0x02	0xFF	0xFF	XX	XX

Alan Açıklamaları

HEADER (0xAA): Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x08): RFID blok okuma işlemini belirler.

KEY_TYPE (0x01 - 0x04): Kullanılacak kimlik doğrulama anahtarını belirler.

0x01 → KeyA

0x02 → KeyB

0x03 → Opsiyonel KeyA

0x04 → Opsiyonel KeyB

AUTH_TYPE (0x01 - 0x02): Kimlik doğrulama yöntemini belirler.

0x01 → PICC_AUTHENT1A

0x02 → PICC_AUTHENT1B

START_BLOCK (0xFF): Okunacak ilk blok adresini belirler.

END_BLOCK (0xFF): Okunacak son blok adresini belirler. **Eğer tek bir blok okunacaksa START_BLOCK ve**

END_BLOCK aynı olmalıdır.

CRC_HIGH (XX) & CRC_LOW (XX) (Opsiyonel): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.



Advanced RFID Card Reader Writer

Gelen Mesaj Formatı

Response (RFID Kart Bloklarından Okunan Veri Yanıtı)

Response (Data Response Read from RFID Card Blocks)					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (Block Data - 16 Bytes)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x10	16 Byte Blok Verisi	0xFF	0xFF

Alan Açıklamaları

HEADER (0xAA): Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Yanıtı gönderen cihazın adresini gösterir.

MESSAGE_TYPE (0x10): RFID blok okuma yanıtı olduğunu belirtir.

DATA (Block Data - 16 Bytes): Okunan RFID blok verisini içerir. **Her blok için 16 bayt veri döndürülür.**

CRC_HIGH (0xFF) & CRC_LOW (0xFF) (Opsiyonel): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Bu Komutun Çalışma Prensibi

RFID kartın varlığı kontrol edilir. Eğer kart algılanmazsa, **ERR_CARD_NOT_FOUND** hatası döndürülür.

Başlangıç ve bitiş blokları doğrulanır. Eğer başlangıç bloğu bitiş bloğundan büyükse, **ERR_INVALID_BLOCK_RANGE** hatası döner.

Kimlik doğrulama yapılır.

Eğer doğrulama başarısız olursa, **ERR_AUTHENTICATION_FAILED** hatası döner.

Doğrulama **KeyA**, **KeyB**, Opsiyonel **KeyA** veya Opsiyonel **KeyB** ile yapılır.

Blok okuma işlemi başlatılır.

Eğer **tek bir blok** okunacaksa, sadece **START_BLOCK** okunur.

Eğer **bir blok aralığı** okunacaksa, **START_BLOCK** ile **END_BLOCK** arasındaki tüm bloklar okunur.

Okunan veri yanıt olarak döndürülür.

Her blok için 16 baytlık veri döndürülür.

Olası Hata Kodları	
Hata Kodu	Açıklama
ERR_CARD_NOT_FOUND	Kart algılanamadı, işlem yapılamaz.
ERR_INVALID_BLOCK_RANGE	Geçersiz blok aralığı seçildi.
ERR_AUTHENTICATION_FAILED	Kimlik doğrulama başarısız oldu, yetkilendirme yapılamadı.
ERR_BLOCK_READ_FAILED	Belirtilen bloklardan veri okunamadı.

Özet

Bu komut, RFID kartların belirli bloklarından veri okumak için kullanılır.

Kimlik doğrulama gerektirir (**KeyA**, **KeyB**, Opsiyonel **KeyA**, Opsiyonel **KeyB**).

Eğer blok aralığı belirtilirse, başlangıç ve bitiş blokları arasındaki tüm bloklar okunur.

Başarılı okuma işlemi sonrası, her blok için 16 baytlık veri yanıt olarak döndürülür.

Kimlik doğrulama hatası, geçersiz blok aralığı veya kart algılama hatası gibi durumlar işlem başarısızlığına neden olabilir.

Bu komut, RFID kartlar üzerinde veri okuma işlemlerini yönetmek için önemli bir rol oynar ve güvenli erişim gerektiren uygulamalarda kullanılır.



Advanced RFID Card Reader Writer

7.3.3.2.1 RFID Kart Bloklarından Veri Okuma Komutu Örneği-1

Bu örnek, RFID kartın **0x0F** numaralı blokundan veri okuma işlemini göstermektedir. Kimlik doğrulama için **KeyA** kullanılmış ve okunan veri **HEX formatında** döndürülmüştür.

Command Sent (Command to Read Data from RFID Card Blocks)								
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	KEY_TYPE	AUTH_TYPE	START_BLOCK	END_BLOCK	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x08	0x01	0x02	0x0F	0x0F	0x79	0x07

Açıklamalar:

- HEADER (0xAA):** Komutun başlangıcını belirler. Sabit bir değerdir.
- DEVICE_ADDRESS (0x01):** Komutun gönderildiği cihazın adresidir.
- COMMAND_TYPE (0x08):** RFID kart blok okuma komutudur.
- KEY_TYPE (0x01):** KeyA kullanılarak kimlik doğrulama yapılacağını belirtir.
- AUTH_TYPE (0x02):** PICC_AUTHENT1B kimlik doğrulama yöntemini belirler.
- START_BLOCK (0x0F):** Okunacak ilk blok adresini belirtir.
- END_BLOCK (0x0F):** Okunacak son blok adresini belirtir (tek bir blok okuma).
- CRC_HIGH (0x79) & CRC_LOW (0x07):** Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response (RFID Card Block Read Response - Block 0x0F)					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (Block 0x0F - 16 Bytes)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x10	0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF	0xBD	0x24

Açıklamalar:

- HEADER (0xAA):** Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.
- DEVICE_ADDRESS (0x01):** Yanıtı gönderen cihazın adresini gösterir.
- MESSAGE_TYPE (0x10):** RFID kart blok okuma işleminden dönen veri mesajıdır.
- DATA:** 0x0F blok adresinden okunan 16 baytlık veridir.

7.3.3.2.2 RFID Kart Bloklarından Veri Okuma Komutu Örneği-2

Bu örnek, RFID kartın 0x01 bloğundan veri okuma işlemini göstermektedir.

Kimlik doğrulama için KeyA kullanılmış ve okunan veri HEX formatında döndürülmüştür.

Gönderilen Mesaj Formatı

Command Sent (Command to Read Data from RFID Card Blocks)								
HEADER	DEVICE _ADDRESS	COMMAND _TYPE	KEY _TYPE	AUTH _TYPE	START _BLOCK	END _BLOCK	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x08	0x01	0x02	0x01	0x01	0xBB	0xC6

Açıklamalar:

HEADER (0xAA): Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x08): RFID kart blok okuma komutudur.

KEY_TYPE (0x01): KeyA kullanılarak kimlik doğrulama yapılacağını belirtir.

AUTH_TYPE (0x02): PICC_AUTHENT1B kimlik doğrulama yöntemini belirler.

START_BLOCK (0x01): Okunacak ilk blok adresini belirtir.

END_BLOCK (0x01): Okunacak son blok adresini belirtir (tek bir blok okuma).

CRC_HIGH (0xBB) & CRC_LOW (0xC6): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response (RFID Card Block Read Response - Block 0x01)					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (Block 0x01 - 16 Bytes)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x10	0x52 0x46 0x49 0x44 0x41 0x58 0x20 0x44 0x65 0x76 0x69 0x63 0x65 0x73 0x00 0x00	0x97	0xF3

Açıklamalar:

HEADER (0xAA): Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Yanıtı gönderen cihazın adresini gösterir.

MESSAGE_TYPE (0x10): RFID kart blok okuma işleminden dönen veri mesajıdır.

DATA: 0x01 blok adresinden okunan 16 baytlık veridir.



Advanced RFID Card Reader Writer

Okunan Blok 0x01 İçeriği

Byte Index	HEX Değeri	ASCII Karşılığı
0	0x52	R
1	0x46	F
2	0x49	I
3	0x44	D
4	0x41	A
5	0x58	X
6	0x20	(Boşluk)
7	0x44	D
8	0x65	e
9	0x76	v
10	0x69	i
11	0x63	c
12	0x65	e
13	0x73	s
14	0x00	NULL
15	0x00	NULL

Ek Açıklamalar:

İlk 14 bayt (0x52 0x46 0x49 0x44 0x41 0x58 0x20 0x44 0x65 0x76 0x69 0x63 0x65 0x73) → ASCII formatında "RFIDAX Devices" yazısını içermektedir.

Son 2 bayt (0x00 0x00) → Boş veri veya blok sonlandırıcı olarak kullanılmıştır.

Özet

Bu örnekte, RFID kartın 0x01 numaralı bloğundan 16 bayt veri okunmuştur.

Cihaz, okuma işlemini başarıyla tamamlamış ve ASCII formatında "RFIDAX Devices" metnini içeren 16 baytlık blok verisini döndürmüştür.

CRC değerleri (0x97 0xF3), mesajın doğruluğunu kontrol etmek için kullanılmaktadır.

Bu mesaj formatı, RFID blok okuma işlemlerinin nasıl yapıldığını anlamak için önemli bir referans niteliğindedir.

Not: RFIDAX cihazlarında okuma ve yazma işlemi sadece anlamlı bloklarda olur. Özel sektör bloklarında bu işlemlere dikkat edilmez. Bu özel bloklara yazma işlemini bilerek kapatılmıştır. RFID kartların bozulmaması için bu işlem yapılmıştır.

7.3.3.2.3 RFID Kart Bloklarından Veri Okuma Komutu Örneği- 3

Bu örnek, RFID kartın belirli bir blok aralığından veri okuma işlemini göstermektedir.

Kimlik doğrulama (KeyA, KeyB, Opsiyonel_KeyA, Opsiyonel_KeyB) ile başlangıç ve bitiş blok adresleri belirlenerek HEX formatında veri okunmaktadır.

Gönderilen Mesaj Formatı

Command Sent (Command to Read Data from RFID Card Blocks)								
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	KEY_TYPE	AUTH_TYPE	START_BLOCK	END_BLOCK	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x08	0x01	0x02	0x10	0x12	0xB9	0xF7

Gelen Mesaj Formatı

Response (RFID Card Block Read Response)

Aşağıdaki mesaj, belirtilen blok aralığındaki verileri döndürmektedir.

HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (Block Contents)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x10	0x46 0x6F 0x72 0x20 0x69 0x6E 0x66 0x6F 0x72 0x6D 0x61 0x74 0x69 0x6F 0x6E 0x2C	0xEA	0xF5
0xAA	0x01	0x10	0x20 0x76 0x69 0x73 0x69 0x74 0x20 0x77 0x77 0x77 0x2E 0x72 0x66 0x69 0x64 0x61	0xD6	0xEA
0xAA	0x01	0x10	0x78 0x2E 0x63 0x6F 0x6D 0x2E 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00	0xD5	0x5A
0xAA	0x01	0x10	0x00 0x00 0x00 0x00 0x00 0x00 0xFF 0x07 0x80 0x69 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF	0xBD	0x24

Okunan Blok İçeriği

Aşağıdaki tablo, her blok içeriğini ayrıntılı olarak göstermektedir.

Blok Adresi	HEX Değerleri	ASCII Karşılığı
0x10	0x46 0x6F 0x72 0x20 0x69 0x6E 0x66 0x6F 0x72 0x6D 0x61 0x74 0x69 0x6F 0x6E 0x2C	For information,
0x11	0x20 0x76 0x69 0x73 0x69 0x74 0x20 0x77 0x77 0x77 0x2E 0x72 0x66 0x69 0x64 0x61	visit www.rfida
0x12	0x78 0x2E 0x63 0x6F 0x6D 0x2E 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00	x.com.

Alan Açıklamaları

HEADER (0xAA) → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresini gösterir.

MESSAGE_TYPE (0x10) → RFID blok okuma mesajı olduğunu belirtir.

DATA → Okunan blok verilerini içerir.

CRC_HIGH & CRC_LOW → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Bu örnekte, RFID kartın 0x10 - 0x12 blokları arasındaki veri başarıyla okunmuş ve ASCII formatında "For information, visit www.rfidax.com." metni elde edilmiştir.

Özet

Bu örnekte, RFID kartın 0x10 - 0x12 blok aralığından veri başarıyla okundu.

Elde edilen ASCII metni: "For information, visit www.rfidax.com."

Her blok 16 bayt veri içerir ve mesaj birden fazla bloğa bölünmüştür.

CRC değerleri, mesaj bütünlüğünü sağlamak ve verinin doğruluğunu kontrol etmek için kullanılmıştır.

Bu mesaj formatı, **çoklu blok okuma senaryolarında** RFID blok okuma işlemlerini anlamak için önemli bir referans niteliğindedir.

7.4 Sistem Kontrol Komutları

RFIDAX cihazlarında **sistem kontrol komutları**, cihazın donanım ve yazılım yönetimi ile ilgili işlemleri gerçekleştirmek için kullanılır. Bu komutlar, cihazın **çalışma durumunu sorgulama, versiyon bilgilerini alma, sistemin yeniden başlatılması ve hata tespiti** gibi önemli işlemleri kapsar.

Cihazın **donanım ve yazılım versiyon numaraları, yapı numarası (build number), üretim tarihi, çalışma modu ve iletişim protokolleri** gibi kritik bilgileri bu komutlarla sorgulanabilir. Ayrıca, cihazın **yeniden başlatılması veya sistem sıfırlaması** gibi işlemler de sistem kontrol komutları üzerinden gerçekleştirilebilir.

Bu komutlar, özellikle **cihazın bakım süreçleri, sistem güncellemeleri ve hata ayıklama işlemlerinde** büyük kolaylık sağlamaktadır. Kullanıcılar ve geliştiriciler, bu komutları kullanarak cihazın sistemsel durumu hakkında bilgi edinebilir, sistemin belirli yapılandırmalarını doğrulayabilir ve gerektiğinde cihazın yeniden başlatılmasını sağlayabilir.

7.4.1 Sistem Kontrol Komutlarının Kullanım Amaçları

Aşağıdaki başlıca amaçlar doğrultusunda **sistem kontrol komutları** kullanılmaktadır:

- Cihazın Donanım ve Yazılım Bilgilerini Sorgulama**
Cihazın **donanım ve yazılım sürümünü** öğrenmek.
Yapı numarasını (build number) ve üretim tarihini sorgulamak.
- Sistem Yeniden Başlatma ve Sıfırlama İşlemleri**
Cihazı yazılım seviyesinde sıfırlamak ve yeniden başlatmak.
Belirli durumlarda cihazın **fabrika ayarlarına geri döndürülmesini** sağlamak.
- Çalışma Modlarını ve Konfigürasyonu Görüntüleme**
Cihazın mevcut iletişim modunu (USB, RS485, vb.) öğrenmek.
Veri bütünlüğü kontrolü (CRC) aktif mi, değil mi sorgulamak.
Cihazın çalışma modunu (Basic, Advanced, Enterprise, Custom) doğrulamak.

7.4.2 Sistem Kontrol Komutları ve Listesi

Aşağıda RFIDAX sisteminde kullanılan temel **sistem kontrol komutları** listelenmiştir:

Komut Kodu	Komut Adı	Açıklama
0x0A	Yazılım Sıfırlama Komutu	Cihazı yazılım seviyesinde yeniden başlatır. Bu işlem, cihazı fabrika ayarlarına döndürmez ancak tüm geçici belleği temizler ve sistemi baştan başlatır.
0x0D	Cihaz Versiyon Bilgisi Alma Komutu	Cihazın donanım ve yazılım versiyon bilgilerini, yapı numarasını ve üretim tarihini döndürür. Sistem üzerindeki mevcut yapılandırmaları öğrenmek için kullanılır.

Tablo 36 Sistem kontrol komutları

Bu komutlar, **cihazın genel yönetimi ve bakım işlemlerinde** büyük önem taşımaktadır.

7.4.2 1 Yazılım Sıfırlama Komutu

Cihazın yazılım seviyesinde yeniden başlatılmasını sağlayan **Yazılım Sıfırlama Komutu (0x0A)**, sistemin **geçici verilerini temizleyerek yeniden başlatılmasını sağlar**.

Bu işlem, cihazın donanımsal olarak sıfırlanmasını gerektirmeden **hafızayı temizler ve işletim döngüsünü yeniden başlatır**. Ancak, cihazın fabrika ayarlarına dönmelerini sağlamaz ve kalıcı veriler etkilenmez.

Bu komut Gereklikli sebebi:

Yazılım sıfırlama, RFID sistemlerinde **hata ayıklama, sistem kararlılığını sağlama ve yazılım güncellemeleri sonrasında cihazı yeniden başlatma** gibi durumlarda kritik bir rol oynar.

Cihazın fiziksel olarak sıfırlanması her zaman mümkün olmayabilir. Özellikle uzak sistemlerde veya gömülü sistemlerde çalışan cihazlarda, uzaktan yönetilebilir bir sıfırlama mekanizmasına ihtiyaç duyulur.

Bu komut, aşağıdaki durumlarda kullanışlıdır:

Hafıza veya işlemci yükünün fazla olduğu durumlarda, sistemin sorunsuz çalışmasını sağlamak için hafızanın temizlenmesi.

Cihazın yazılım güncellemesi veya konfigürasyon değişikliği sonrası, yeni ayarların geçerli olması için yeniden başlatılması.

Beklenmeyen bir hata veya kilitlenme durumunda, donanımsal sıfırlama gerektirmeden sistemin yazılım seviyesinde yeniden başlatılması.

Test ve geliştirme aşamalarında, yeni firmware yüklenmeden önce cihazın resetlenerek sistemin temizlenmesi.

Bu sayede, **manuel müdahaleye gerek kalmadan sistemin stabil çalışması sağlanır** ve cihazın çalışmasını sürdürmesine olanak tanır.

Gönderilen Mesaj Formatı

Command Sent (Software Reset Command for the Device)					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0A	0xFF	0xC0	0x16

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x0A) → Yazılım sıfırlama işlemini başlatır.

SUB_COMMAND_TYPE (0xFF) → Sıfırlama türünü belirten alt komut.

CRC_HIGH (0xC0) & CRC_LOW (0x16) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Bu komut başarıyla işlendiğinde, cihaz kendini sıfırlar ve yeniden başlar. Eğer komut geçersizse veya sistem sıfırlama işlemi başarısız olursa, cihaz hata kodu ile yanıt dönebilir. Lütfen bir hata ile karşılaşırsanız "[6.4.1 Genel Sistem Durumu ve Hataları](#)" bakınız.

Bu işlem sonucunda, herhangi bir ayarlama (setup) işlemi sonrasında veya bir reset durumu sonrası aşağıdaki ASCII mesajı geri döner. Bu mesaj ASCII formatındadır.

Advanced RFID System Initialization(Örnektir)

[Communication Mode]: Mobile (Type-C)

[Data Integrity Check]: Enabled (CRC Active)

[Operation Mode]: Advanced

[Hardware Version]: 1.2

[Software Version]: 3.2

[Build Number]: 75

[Build Date]: 202501

```
=====
Advanced RFID System Initialization
=====
[Communication Mode]: Mobile (Type-C)
[Data Integrity Check]: Enabled (CRC Active)
[Operation Mode]: Advanced
=====
[Hardware Version]: 1.2
[Software Version]: 3.2
[Build Number]: 75
[Build Date]: 202501
=====
```

Şekil 3 Reset sonrası ASCII mesajı (Örnektir)

7.4.2.2 Cihaz Versiyon Bilgisi Alma Komutu

Device_Version_Info, bir cihazın donanım ve yazılım sürüm bilgilerini, yapı numarasını, üretim tarihini ve mevcut çalışma modlarını belirlemek için kullanılan kritik bir komuttur. Bu komut sayesinde, cihazın güncel versiyon bilgileri sistem tarafından okunabilir ve bu bilgiler yazılım uyumluluğu, güncellemeler ve hata ayıklama süreçlerinde önemli bir rol oynar.

Sistem geliştiricileri ve bakım ekipleri için, **Device_Version_Info** komutunun kullanımı şu avantajları sağlar:

- Uyumluluk Kontrolü:**
Donanım ve yazılım sürümlerinin eşleşmesini doğrular.
Farklı sürümlere sahip cihazların doğru çalışmasını sağlar.
- Hata Ayıklama ve Destek:**
Destek ekibi, sürüm bilgilerini kullanarak hata kaynaklarını hızlı bir şekilde belirleyebilir.
Cihazın geçmiş güncellemelerini takip etmek için kullanılabilir.
- Güncellemelerin Yönetimi:**
Hangi cihazların güncellenmesi gerektiğini belirler.
Güncelleme sonrası versiyon kontrolü yapılmasını sağlar.
- Cihaz Konfigürasyonlarının Doğrulanması:**
Mevcut iletişim modu (**CommMode**), CRC denetim durumu (**CRCMode**) ve çalışma modu (**OperationMode**) bilgileri sayesinde cihazın yapılandırması okunabilir.

Bu bilgiler, özellikle büyük ölçekli sistemlerde, farklı versiyonlara sahip cihazların doğru yönetimi için hayati önem taşır. **Device_Version_Info**, sistemin kararlılığını sağlamak ve potansiyel uyumluluk sorunlarını önlemek için düzenli olarak kullanılmalıdır.

7.4.2.2.1 Cihaz Versiyon Bilgisi Alma Komutu Genel Mesaj Yapısı

Cihaz versiyon bilgisini almak için kullanılan mesaj yapısı, **gönderilen komut (Command Sent)** ve **gelen yanıt (Response Message)** olmak üzere iki ana bileşenden oluşur:

- Gönderilen Komut:**
Cihaza versiyon bilgilerini döndürmesi için gönderilen istek mesajıdır.
İçerisinde **SUB_COMMAND_TYPE** bilgisi bulunur, ancak veri içeriği (DATA) yoktur.
CRC hesaplaması, mesaj bütünlüğünü doğrulamak için eklenmiştir.
- Gelen Yanıt:**
Cihazdan alınan versiyon bilgilerini içeren yanıt mesajıdır.
Donanım ve yazılım sürümü, yapı numarası, üretim tarihi ve mevcut çalışma modlarını içerir.
CRC değeri, mesajın bütünlüğünü sağlamak amacıyla eklenmiştir.

Gönderilen Mesaj Formatı

Command Sent (Cihaz Versiyon Bilgisi Okuma Komutu)					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0A	0xFF	XX	XX

Alan Açıklamaları:

- HEADER (0xAA):** Komutun başlangıcını belirler. Sabit bir değerdir.
- DEVICE_ADDRESS (0x01):** Komutun gönderildiği cihazın adresidir.
- COMMAND_TYPE (0x0A):** Cihaz versiyon bilgisi talep etme komutudur.
- SUB_COMMAND_TYPE (0xFF):** Cihaz versiyon bilgisi okuma alt komutudur.
- CRC_HIGH & CRC_LOW:** Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response Message (Cihaz Versiyon Bilgisi Yanıtı)											
HEADER	DEVICE _ADDRESS	MESSAGE_TYPE	Hardware Ver.	Software Ver.	Build No.	Build Date	Comm Mode	CRC Mode	Operation Mode	CRC_HIGH	CRC_LOW
0xAA	0x01	0x02	0xFF 0xFF	0xFF 0xFF	0xFF 0xFF 0xFF 0xFF 0xFF	0xFF 0xFF 0xFF 0xFF 0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

Yanıt Açıklamaları:

- HEADER (0xAA):** Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.
- DEVICE_ADDRESS (0x01):** Yanıtı gönderen cihazın adresini gösterir.
- MESSAGE_TYPE (0x02):** Cihaz versiyon bilgisinin gönderildiğini belirtir.
- Hardware Version (0xFF 0xFF):**
 - Major:** Cihazın donanım ana sürüm numarası
 - Minor:** Cihazın donanım alt sürüm numarası
- Software Version (0xFF 0xFF):**
 - Major:** Cihazın yazılım ana sürüm numarası
 - Minor:** Cihazın yazılım alt sürüm numarası
- Build Number (0xFF 0xFF 0xFF 0xFF):** Cihazın mevcut yazılım derleme numarası
- Build Date (0xFF 0xFF 0xFF 0xFF 0xFF):** Cihazın üretim tarihi veya son yazılım güncelleme tarihi
- Communication Mode (0xFF):** Mevcut iletişim modu. Detaylı bilgi için bakınız. ([6.3 Cihazların Haberleşme Türleri](#))
- CRC Mode (0xFF):** Veri bütünlüğü kontrol durumu (CRC açık veya kapalı). Detaylı bilgi için bakınız. ([6.2 Veri Bütünlüğü Modları ve Açıklamaları](#))
- Operation Mode (0xFF):** Cihazın çalışma modu. Detaylı bilgi için bakınız. ([6.1 Cihazın Modları ve Açıklamaları](#))
- CRC_HIGH & CRC_LOW:** Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

7.4.2 2.1.1 Cihaz Versiyon Bilgisi Alma Komutu Genel Mesaj Yapısı Örneği 1

AA 01 02 00 01 00 02 00 03 00 02 00 00 00 4B 32 30 32 35 30 31 03 01 01 00 F7 EC

Gelen Yanıt (Örnek Veri)											
HEADER	DEVICE _ADDRESS	MESSAGE _TYPE	Hardware Ver.	Software Ver.	Build No.	Build Date	Comm Mode	CRCMode	Operation Mode	CRC_HIGH	CRC_LOW
0xAA	0x01	0x02	0x00 0x01 0x00 0x02	0x00 0x03 0x00 0x02	0x00 0x00 0x00 0x4B	0x32 0x30 0x32 0x35 0x30 0x31	0x03	0x01	0x01	0xF7	0xEC

Yanıt Mesajı Açıklaması

BYTE INDEX	HEX DEĞERİ	AÇIKLAMA
0	0xAA	HEADER: Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.
1	0x01	DEVICE_ADDRESS: Yanıtı gönderen cihazın adresini gösterir.
2	0x02	MESSAGE_TYPE: Cihaz versiyon bilgisi yanıtı olduğunu belirtir.
3-4	0x00 0x01	Hardware Version: Donanım sürüm numarası (1.2)
5-6	0x00 0x02	Software Version: Yazılım sürüm numarası (3.2)
7-10	0x00 0x00 0x00 0x4B	Build Number: Cihazın derleme numarası (75)
11-16	0x32 0x30 0x32 0x35 0x30 0x31	Build Date (YYYYM Format): 202501 (Ocak 2025)
17	0x03	Communication Mode: Cihazın iletişim modu (COMM_TYPE_C)
18	0x01	CRC Mode: Veri bütünlüğü kontrolü etkin (0x01)
19	0x01	Operation Mode: Gelişmiş Mod (MODE_ADVANCED) etkin
20	0xF7	CRC_HIGH: Veri bütünlüğünü sağlayan hata kontrol mekanizmasının ilk baytı.
21	0xEC	CRC_LOW: Veri bütünlüğünü sağlayan hata kontrol mekanizmasının ikinci baytı.

Özet

Bu mesaj, cihazın donanım ve yazılım versiyon bilgilerini, **build numarasını**, **üretim tarihini** ve **çalışma modlarını** içeren önemli bir yanıt mesajıdır.

Donanım ve yazılım sürümleri: 1.2 (HW) ve 3.2 (SW) olarak belirlenmiştir.

Build numarası: 75 olarak belirlenmiştir.

Üretim tarihi: Ocak 2025 (202501) olarak kodlanmıştır.

İletişim modu: 0x03 → COMM_TYPE_C

CRC kontrolü: 0x01 → Açık, mesaj bütünlüğü doğrulanıyor.

Çalışma modu: 0x01 → Gelişmiş Mod (MODE_ADVANCED).

Bu yapı, cihazın **güncel versiyon bilgilerini** almak, yazılım uyumluluğunu kontrol etmek ve hata giderme işlemlerinde kullanılmak üzere tasarlanmıştır.

7.5 Bellek Yönetim Komutları

RFIDAX cihazlarında **Bellek Yönetim Komutları**, cihazın güvenlik anahtarlarını (optionalKeyA ve optionalKeyB), kimlik bilgilerini (ID) ve belirli hafıza bölümlerini yönetmek için kullanılır. Bu komutlar, cihazın yetkilendirilmiş kullanıcıları tarafından belirlenen güvenlik seviyelerine uygun şekilde çalışır.

Bellek yönetimi, cihazın güvenlik yapılandırmalarını ve kullanıcı bilgilerini içeren özel bir sistemdir. Bu yapı sayesinde, **kimlik doğrulama anahtarları** güvenli bir şekilde saklanabilir ve güncellenebilir. Ayrıca, **ID yönetimi** komutları sayesinde, RFID cihazının benzersiz kimlik bilgileri okunabilir, değiştirilebilir veya sıfırlanabilir.

Bu komutlar, **ileri seviye erişim kontrolü ve güvenlik işlemleri** için tasarlanmıştır. RFID cihazının kullanıcılar, sistem yöneticileri veya güvenlik protokolleri tarafından doğru ve güvenilir bir şekilde yönetilmesine olanak tanır.

7.5.1 Bellek Yönetim Komutlarının Kullanım Amaçları

Bellek yönetim komutları, aşağıdaki amaçlar doğrultusunda kullanılmaktadır:

Güvenlik Anahtarlarını Yönetme

- OptionalKeyA ve optionalKeyB değerlerini kaydetme ve okuma işlemleri
- Kimlik doğrulama süreçlerini yönetme

ID (Kimlik Bilgisi) İşlemleri

- Cihazın **benzersiz kimlik numarasını (ID)** güncelleme
- Mevcut **ID bilgilerini okuma ve sıfırlama**
- Cihaza kayıtlı **tüm ID'leri listeleme**

Bu komutlar, cihazın **veri güvenliğini, erişim yönetimini ve kimlik doğrulama sistemlerini** daha güçlü hale getirmek için kritik bir rol oynamaktadır.

Aşağıda, RFIDAX cihazlarında kullanılan **Bellek Yönetim Komutları** listelenmiştir:

Komut Kodu	Komut Adı	Açıklama
0x0B	OptionalKeyA ve optionalKeyB Yönetimi	Güvenlik anahtarlarını kaydetme ve okuma işlemleri
0x0C	ID Yönetimi	Benzersiz kimlik (ID) işlemleri

Tablo 37 Bellek Yönetim Komutları

0x0B - optionalKeyA ve optionalKeyB İşlemleri

Alt Komut Kodu	İşlem
0x0B / 0x01	OptionalKeyA yazma
0x0B / 0x02	OptionalKeyB yazma
0x0B / 0x03	OptionalKeyA okuma
0x0B / 0x04	OptionalKeyB okuma

Tablo 38 OptionalKeyA ve optionalKeyB İşlemleri

0x0C - ID İşlemleri

Alt Komut Kodu	İşlem
0x0C / 0x01	ID güncelleme
0x0C / 0x02	ID okuma
0x0C / 0x03	ID sıfırlama
0x0C / 0x04	Tüm ID'leri gösterme

Tablo 39 ID İşlemleri



Advanced RFID Card Reader Writer

7.5.2 OptionalKeyA ve OptionalKeyB İşlemleri

RFIDAX cihazlarında **OptionalKeyA** ve **OptionalKeyB**, kullanıcı tarafından belirlenebilen kimlik doğrulama anahtarlarıdır. Bu anahtarlar, standart **KeyA** ve **KeyB**'den farklı olarak, **yazılabilir ve okunabilir** niteliktedir. Güvenliği artırmak ve özelleştirilebilir erişim yönetimi sağlamak amacıyla kullanılmaktadır.

7.5.2.1 OptionalKeyA ve OptionalKeyB Komutlarının Kullanım Amaçları

Optional Key işlemleri, RFID kartların **güvenli erişimini yönetmek** için kullanılır.

Bu komutlarla aşağıdaki işlemler gerçekleştirilebilir:

Optional KeyA ve Optional KeyB Güncelleme (Yazma)

Yeni bir **OptionalKeyA** veya **OptionalKeyB** değeri belirlenerek **Flash Bellek** üzerinde saklanabilir.

Yetkili kullanıcılar tarafından anahtar güncellenebilir.

Optional KeyA ve Optional KeyB Okuma

Daha önce kaydedilmiş **OptionalKeyA** veya **OptionalKeyB** anahtarları okunabilir.

Cihazın mevcut kimlik doğrulama ayarları kontrol edilebilir.

7.5.2.2 Gönderilen Mesaj Formatı

Command Sent (Command to Read or Write OptionalKeyA / OptionalKeyB)						
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	KEY_TYPE	DATA (16 Bytes - Optional)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0B	0xFF	(Sadece Yazma Komutlarında)	0xFF	0xFF

Alan Açıklamaları

HEADER (0xAA): Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x0B): OptionalKeyA veya OptionalKeyB işlemlerini belirler.

KEY_TYPE (0xFF): Yapılacak işlemi belirler:

0x01 → OptionalKeyA yazma

0x02 → OptionalKeyB yazma

0x03 → OptionalKeyA okuma

0x04 → OptionalKeyB okuma

DATA (16 Bytes - Optional):

Yazma komutlarında, yeni anahtar verisini içerir.

Okuma komutlarında, bu alan kullanılmaz.

CRC_HIGH (XX) & CRC_LOW (XX) (Opsiyonel): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

7.5.2.3 Gelen Mesaj Formatı (Yazma İşlemi Sonrası)

Response (Optional Key Write Response)						
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (Optional)	CRC_LOW (Optional)	
0xAA	0x01	0x00	0x00	0xFF	0xFF	

Alan Açıklamaları

HEADER (0xAA): Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Yanıtı gönderen cihazın adresini gösterir.

STATUS_CODE_HIGH (0x00) & STATUS_CODE_LOW (0x00):

0x00 0x00 → Başarıyla yazıldı.

Başka bir hata kodu: Yazma işlemi başarısız oldu. Hata koduna göre buraya” [6.4.1 Genel Sistem Durumu ve Hataları](#)”bakınız

CRC_HIGH (XX) & CRC_LOW (XX) (Opsiyonel): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

7.5.2.4 Gelen Mesaj Formatı (Okuma İşlemi Sonrası)

Response (Optional Key Read Response)

HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (16 Bytes - Key Value)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x11 (KeyA) / 12 (KeyB)	16 Byte Key Data	0xFF	0xFF

Alan Açıklamaları

HEADER (0xAA): Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Yanıtı gönderen cihazın adresini gösterir.

MESSAGE_TYPE:

0x11 → OptionalKeyA'nın değerini gösterir.

0x12 → OptionalKeyB'nin değerini gösterir.

DATA (16 Bytes): Okunan **OptionalKeyA** veya **OptionalKeyB** değerini içerir.

CRC_HIGH (XX) & CRC_LOW (XX) (Opsiyonel): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

7.5.2.5 OptionalKeyA Yazma Komutu Örneği

Bu örnek, **OptionalKeyA** değerini güncellemek için kullanılan komutun yapısını ve işlem sürecini göstermektedir.

Gönderilen Mesaj Formatı

Command Sent (Command to Write OptionalKeyA to RFID System)						
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	KEY_TYPE	DATA (16 Bytes - New Key Value)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0B	0x01	0x1A 0x2A 0x3A 0x4A 0x5A 0x6A	0x4D	0x06

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x0B) → OptionalKey işlemleri komutunu belirtir.

KEY_TYPE (0x01) → **OptionalKeyA yazma işlemi**ni belirtir.

DATA (16 Bytes - New Key Value) → Yeni **OptionalKeyA** değeri olarak 16 baytlık veri içermektedir.

0x1A 0x2A 0x3A 0x4A 0x5A 0x6A → Yeni KeyA verisi

CRC_HIGH (0x4D) & CRC_LOW (0x06) (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response (OptionalKeyA Write Response)						
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (Optional)	CRC_LOW (Optional)	
0xBB	0x01	0x00	0x00	0x5C	0x3E	

Alan Açıklamaları

HEADER (0xBB) → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresini gösterir.

STATUS_CODE_HIGH (0x00) & STATUS_CODE_LOW (0x00) →

0x00 0x00 → Başarıyla yazıldı.

Farklı bir hata kodu → Yazma işlemi başarısız oldu.

CRC_HIGH (0x5C) & CRC_LOW (0x3E) (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

7.5.2.6 OptionalKeyA Okuma Komutu Örneği

Bu örnek, **OptionalKeyA** değerinin okunması için kullanılan komutun yapısını ve işlem sürecini göstermektedir. **OptionalKeyA**, RFID kimlik doğrulama süreçlerinde kullanılan özel bir anahtardır. Kullanıcılar, bu komut ile cihazın mevcut **OptionalKeyA** değerini okuyabilirler.

Gönderilen Mesaj Formatı

Command Sent (Command to Read OptionalKeyA from RFID System)					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	KEY_TYPE	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0B	0x03	0xDD	0xB4

Alan Açıklamaları

- HEADER (0xAA)** → Komutun başlangıcını belirler. Sabit bir değerdir.
- DEVICE_ADDRESS (0x01)** → Komutun gönderildiği cihazın adresidir.
- COMMAND_TYPE (0x0B)** → OptionalKey işlemleri komutunu belirtir.
- KEY_TYPE (0x03)** → OptionalKeyA okuma işlemini belirtir.
- CRC_HIGH (0xDD) & CRC_LOW (0xB4) (Opsiyonel)** → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response (OptionalKeyA Read Response)					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (OptionalKeyA - 6 Bytes)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x11	0x1A 0x2A 0x3A 0x4A 0x5A 0x6A	0x47	0x47

Alan Açıklamaları

- HEADER (0xAA)** → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.
- DEVICE_ADDRESS (0x01)** → Yanıtı gönderen cihazın adresini gösterir.
- MESSAGE_TYPE (0x11)** → OptionalKeyA okuma mesajı olduğunu belirtir.
- DATA (OptionalKeyA - 6 Bytes)** → Cihazdan okunan mevcut **OptionalKeyA** değeridir.
- 0x1A 0x2A 0x3A 0x4A 0x5A 0x6A** → OptionalKeyA değeri.
- CRC_HIGH (0x47) & CRC_LOW (0x47) (Opsiyonel)** → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Bu İşlemden Gerçekleşen Adımlar

- Kullanıcı, OptionalKeyA değerini okumak için ilgili komutu cihazına gönderir.
- Cihaz, OptionalKeyA değerini belleğinden alır.
- Mevcut OptionalKeyA değeri, 6 baytlık veri olarak yanıt mesajında döndürülür.
- Cihaz, mesajın doğruluğunu sağlamak için CRC değerlerini ekleyerek yanıtı tamamlar.



Advanced RFID Card Reader Writer

7.5.2.7 OptionalKeyB Yazma Komutu Örneği

Bu örnek, **OptionalKeyB** değerinin güncellenmesi için kullanılan komutun yapısını ve işlem sürecini göstermektedir. **OptionalKeyB**, RFID kimlik doğrulama süreçlerinde kullanılan özel bir anahtar olup, kullanıcılar bu komut ile **OptionalKeyB** değerini değiştirebilirler.

Gönderilen Mesaj Formatı

Command Sent (Command to Write OptionalKeyB to RFID System)						
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	KEY_TYPE	DATA (OptionalKeyB - 6 Bytes)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0B	0x02	0xA1 0xB2 0x99 0xD1 0xE1 0xF1	0x80	0x6A

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x0B) → OptionalKey işlemleri komutunu belirtir.

KEY_TYPE (0x02) → OptionalKeyB yazma işlemini belirtir.

DATA (OptionalKeyB - 6 Bytes) → Yeni OptionalKeyB değeri.

0xA1 0xB2 0x99 0xD1 0xE1 0xF1 → Kullanıcı tarafından belirlenen yeni OptionalKeyB değeri.

CRC_HIGH (0x80) & CRC_LOW (0x6A) (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response (OptionalKeyB Write Response)					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xBB	0x01	0x00	0x00	0x5C	0x3E

Alan Açıklamaları

HEADER (0xBB) → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresini gösterir.

STATUS_CODE_HIGH (0x00) & STATUS_CODE_LOW (0x00) → Başarılı yazma işlemi olduğunu gösterir.

CRC_HIGH (0x5C) & CRC_LOW (0x3E) (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

7.5.2.8 OptionalKeyB Okuma Komutu Örneği

Bu örnek, **OptionalKeyB** değerinin okunması için kullanılan komutun yapısını ve işlem sürecini göstermektedir. **OptionalKeyB**, RFID kimlik doğrulama süreçlerinde kullanılan özel bir anahtardır. Kullanıcılar, bu komut ile cihazda kayıtlı olan **OptionalKeyB** değerini sorgulayabilirler.

Gönderilen Mesaj Formatı

Command Sent (Command to Read OptionalKeyB from RFID System)					
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	KEY_TYPE	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0B	0x04	0xAD	0x53

Alan Açıklamaları

- HEADER (0xAA)** → Komutun başlangıcını belirler. Sabit bir değerdir.
DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın adresidir.
COMMAND_TYPE (0x0B) → OptionalKey işlemleri komutunu belirtir.
KEY_TYPE (0x04) → OptionalKeyB okuma işlemini belirtir.
CRC_HIGH (0xAD) & CRC_LOW (0x53) (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response (OptionalKeyB Read Response)					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (OptionalKeyB - 6 Bytes)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x12	0xA1 0xB2 0x99 0xD1 0xE1 0xF1	0x8A	0x2B

Alan Açıklamaları

- HEADER (0xAA)** → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.
DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresini gösterir.
MESSAGE_TYPE (0x12) → RFID sisteminde **OptionalKeyB** değerinin döndürüldüğünü gösterir.
DATA (0xA1 0xB2 0x99 0xD1 0xE1 0xF1) → **OptionalKeyB** değeri olarak kaydedilmiş anahtar.
CRC_HIGH (0x8A) & CRC_LOW (0x2B) (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

7.5.2.9 Özet

OptionalKeyA ve **OptionalKeyB**, gelişmiş kimlik doğrulama ve güvenlik işlemleri için kullanılan özel anahtar değerleridir.

0x0B komut grubu, bu anahtarların yazılması ve okunması işlemlerini yönetir.

Yazma işlemi tamamlandıktan sonra, cihaz bir başarı veya hata yanıtı döndürür.

Okuma işlemi sonucunda, cihaz ilgili anahtarın mevcut değerini döndürür.

Bu bölümde **OptionalKeyA** ve **OptionalKeyB** işlemleri hakkında detaylı bilgi verildi.



Advanced RFID Card Reader Writer

7.5.2 ID İşlemleri

RFIDAX cihazlarında **ID İşlemleri Komutları**, cihazın **benzersiz kimlik (ID) verilerini yönetmek** için kullanılır. **ID yönetimi**, cihazın **tanımlanabilirliğini, kullanıcı erişim yetkilerini ve sistem üzerindeki kimlik doğrulama süreçlerini** yönlendirmek için kritik bir bileşendir.

RFID sistemlerinde **ID bilgisi**, cihazın ya da belirli kullanıcıların benzersiz kimliğini temsil eder. **Bu bilgiler, cihaz belleğinde saklanır ve gerektiğinde güncellenebilir, okunabilir, sıfırlanabilir veya listelenebilir.**

7.5.3.1 ID İşlemlerinin Kullanım Amaçları

ID yönetim komutları, **aşağıdaki işlemleri gerçekleştirmek için** kullanılmaktadır:

ID Güncelleme

Cihazın mevcut ID bilgisi değiştirilebilir veya yeni bir ID atanabilir.

Yetkili kullanıcılar, sistemde önceden belirlenmiş ID numaralarını güncelleyebilir.

ID numaraları, genellikle erişim kontrolü ve cihaz yetkilendirme işlemleri için kullanılır.

ID Okuma

Mevcut ID bilgileri okunabilir.

Cihaz üzerindeki belirli bir ID numarasına ait veri alınabilir.

Yetkilendirme kontrolleri doğrulanarak, kimlik bilgileri sistemde görüntülenebilir.

ID Sıfırlama

Belirli bir ID tamamen sıfırlanabilir ve eski bilgiler silinebilir.

ID sıfırlama işlemi, yanlış yapılandırılmış veya geçersiz ID'leri temizlemek için kullanılır.

Tüm ID'leri Listeleme

Cihazda kayıtlı tüm ID'lerin listesini almak için kullanılır.

Bütün kimlik bilgileri, sistem yöneticileri tarafından incelenebilir ve yönetilebilir.

7.5.3.2 Bellek Yönetimi ve ID İşlemlerinin Önemi

RFIDAX sisteminde **ID işlemleri**, cihazın **tanımlama, yetkilendirme ve güvenlik süreçlerinde** kritik bir rol oynar.

Yetkili kullanıcılar, bu işlemler ile sistemde **ID bazlı erişim yönetimi yapabilir** ve kimlik doğrulama süreçlerini güçlendirebilir.

ID güncelleme ve okuma işlemleri, cihazın **yetkilendirilmiş kişilere özel tanımlanmasını** sağlar.

Gelişmiş kimlik doğrulama mekanizmaları için **doğru ID yönetimi** gereklidir.



Advanced RFID Card Reader Writer

7.5.3.3 Gönderilen Mesaj Formatı

ID yönetimi komutları, cihazın ID verilerini **güncellemek, okumak, sıfırlamak veya tüm ID'leri listelemek** için kullanılır.

Bu komutlar, **ID uzunluğunu ve ID index değerini içeren özel bir formatta** gönderilir.

Hafızada 10 adet ID işlemi yapılabilir.

Command Sent (Command for ID Management Operations)								
HEADER	DEVICE_ ADDRESS	COMMAND_ TYPE	SUB_COMMAND_ TYPE	ID_L ENGT	ID_ INDEX	DATA (Variable)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0C	0xXX	0xXX	0xXX	(Only for Write)	0xXX	0xXX

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın adresini belirtir.

COMMAND_TYPE (0x0C) → ID işlemlerini belirten ana komut türüdür.

SUB_COMMAND_TYPE (0xXX) → Yapılacak işlemi belirler:

0x01 → ID Güncelleme (Update ID)

0x02 → ID Okuma (Read ID)

0x03 → ID Sıfırlama (Reset ID)

0x04 → Tüm ID'leri Listeleme (Show All IDs)

ID_LENGTH (0xXX) → Güncellenecek veya okunacak ID'nin uzunluğu.

ID_INDEX (0xXX) → Güncellenecek veya okunacak ID'nin indeks numarası.

DATA (Variable) → Sadece **ID Güncelleme komutlarında** kullanılır. Yeni ID değerini içerir.

CRC_HIGH (XX) & CRC_LOW (XX) (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

7.5.3.4 Gelen Mesaj Formatı (ID Güncelleme, Okuma ve Sıfırlama Yanıtı)

ID işlemleri için cihazdan gelen yanıt mesajı, işlem sonucunu ve varsa okunan ID verisini içerir.

Response (ID Management Response)					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (Variable - Only for Read)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0xXX	ID Data (Optional)	0xXX	0xXX

Alan Açıklamaları

HEADER (0xAA) → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresini gösterir.

MESSAGE_TYPE (0xXX) → ID işlemi sonucunu belirtir:

0x20+ID(Nummer hex value) → ID Başarıyla Güncellendi

0x50+ID(Nummer hex value) → ID Okuma Sonucu

0x8C +ID(Nummer hex value) → ID Başarıyla Sıfırlandı

0xC8 → Tüm ID'ler Listelendi

DATA (Variable - Only for Read ID & Show All IDs) → Okunan ID verisini içerir.

CRC_HIGH (XX) & CRC_LOW (XX) (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.



Advanced RFID Card Reader Writer

7.5.3.5 ID Güncelleme Komutu Örneği

Bu örnek, RFID cihazında bir ID değerini güncellemek için kullanılan komutun yapısını ve işlem sürecini göstermektedir.

Bu işlem, mevcut ID bilgisini değiştirmek veya yeni bir ID atamak için kullanılır.

1. Örnek: ID 0 Güncelleme İşlemi

Bu işlem, ID index 0'daki değeri güncellemek için kullanılır.

Gönderilen Mesaj Formatı

Command Sent (Command to Update ID 0 in RFID System)								
HEADER	DEVICE _ADDRESS	COMMAN D _TYPE	SUB_COMMAN D _TYPE	ID_ LENGTH	ID _IND EX	DATA (New ID Value)	CRC _HIGH (Optional)	CRC _LOW (Optional)
0xAA	0x01	0x0C	0x01	0x04	0x00	0x12 0x34 0x56 0x78	0x58	0xB9

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler.

DEVICE_ADDRESS (0x01) → Cihazın adresini belirtir.

COMMAND_TYPE (0x0C) → ID işlemlerini belirten ana komuttur.

SUB_COMMAND_TYPE (0x01) → ID güncelleme işlemini belirtir.

ID_LENGTH (0x04) → Güncellenecek ID'nin uzunluğunu (4 byte) gösterir.

ID_INDEX (0x00) → Güncellenecek ID'nin indeksini belirtir.

DATA (0x12 0x34 0x56 0x78) → Yeni ID değeri.

CRC_HIGH (0x58) & CRC_LOW (0xB9) → Hata kontrol mekanizması.

Gelen Mesaj Formatı

Response (ID Update Response for ID 0)					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (Updated ID 0)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x20	0x12 0x34 0x56 0x78	0x3E	0xB7

Alan Açıklamaları

HEADER (0xAA) → Yanıt mesajının başlangıcını belirler.

DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresidir.

MESSAGE_TYPE (0x20) → ID güncelleme işlemini gösterir. Detay için bakınız ([6.7.2.1.2 Flash Bellek ID Güncelleme Mesajları ve Message Type Bilgisi](#))

DATA (0x12 0x34 0x56 0x78) → Güncellenen ID değeri.

CRC_HIGH (0x3E) & CRC_LOW (0xB7) → Hata kontrol mekanizması.

Not : MESSAGE_TYPE bilgisi ID göre değişmektedir. 0x20+ID numarasıdır.



Advanced RFID Card Reader Writer

2. Örnek: ID 9 Güncelleme İşlemi

Bu işlem, ID index 9'daki değeri güncellemek için kullanılır.

Gönderilen Mesaj Formatı

Command Sent (Command to Update ID 9 in RFID System)								
HEADER	DEVICE _ADDRESS	COMMAND _TYPE	SUB_ COMMAND _TYPE	ID _LENGTH	ID _INDEX	DATA (New ID Value)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0C	0x01	0x04	0x09	0x6F 0x6E 0x75 0x72	0xE0	0xC6

Alan Açıklamaları

- HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.
DEVICE_ADDRESS (0x01) → Cihazın adresini belirtir.
COMMAND_TYPE (0x0C) → ID işlemlerini belirten ana komuttur.
SUB_COMMAND_TYPE (0x01) → ID güncelleme işlemini belirtir.
ID_LENGTH (0x04) → Güncellenecek ID'nin uzunluğunu (4 byte) gösterir.
ID_INDEX (0x09) → Güncellenecek ID'nin indeksini belirtir.
DATA (0x6F 0x6E 0x75 0x72) → Yeni ID değeri
CRC_HIGH (0xE0) & CRC_LOW (0xC6) → Hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response (ID Update Response for ID 9)					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (Updated ID 9)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x29	0x6F 0x6E 0x75 0x72	0x86	0xC8

Alan Açıklamaları

- HEADER (0xAA) → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.
DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresidir.
MESSAGE_TYPE (0x29) → ID güncelleme işlemini gösterir (MESSAGE_TYPE = 0x20 + ID_INDEX).
Detay için bakınız [\(6.7.2.1.2 Flash Bellek ID Güncelleme Mesajları ve Message Type Bilgisi\)](#)
DATA (0x6F 0x6E 0x75 0x72) → Güncellenen ID değeri
CRC_HIGH (0x86) & CRC_LOW (0xC8) → Hata kontrol mekanizmasıdır.

Not:

- MESSAGE_TYPE bilgisi, güncellenen ID numarasına bağlıdır.
0x20 + ID_INDEX şeklinde hesaplanır.
ID_INDEX = 9 → MESSAGE_TYPE = 0x20 + 0x09 = 0x29
Her ID güncelleme işleminde, cihaz ilgili ID numarasına göre bir yanıt döndürür.
CRC değerleri, mesajın doğruluğunu sağlamak için kullanılır.



Advanced RFID Card Reader Writer

7.5.3.6 ID Okuma Komutu Örneği

Bu örnek, RFID cihazında bir ID değerinin okunması için kullanılan komutun yapısını ve işlem sürecini göstermektedir.

Bu işlem, mevcut ID bilgisini almak için kullanılır.

1.Örnek: ID 1 Okuma İşlemi

Bu işlem, ID index 1'deki değeri okumak için kullanılır.

Gönderilen Mesaj Formatı

Command Sent (Command to Read ID 1 from RFID System)							
HEADER	DEVICE_ ADDRESS	COMMAND_ TYPE	SUB_COMMAND_ TYPE	ID_ LENGTH	ID_ INDEX	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0C	0x02	0x04	0x01	0x3E	0xDC

Alan Açıklamaları

- HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.
DEVICE_ADDRESS (0x01) → Cihazın adresini belirtir.
COMMAND_TYPE (0x0C) → ID işlemlerini belirten ana komuttur.
SUB_COMMAND_TYPE (0x02) → ID okuma işlemini belirtir.
ID_LENGTH (0x04) → Okunacak ID'nin uzunluğunu (4 byte) gösterir.
ID_INDEX (0x01) → Okunacak ID'nin indeksini belirtir.
CRC_HIGH (0x3E) & CRC_LOW (0xDC) → Hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response (ID Read Response for ID 1)						
HEADER	DEVICE_ ADDRESS	MESSAGE_ TYPE	DATA (Read ID 1)			CRC_HIGH (Optional) CRC_LOW (Optional)
0xAA	0x01	0x51	0x11	0x22	0x33 0x44	0xE0 0x7F

Alan Açıklamaları

- HEADER (0xAA) → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.
DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresidir.
MESSAGE_TYPE (0x51) → ID okuma işlemini gösterir (MESSAGE_TYPE = 0x50 + ID_INDEX).
Detay için bakınız [\(6.7.2.1.3 Flash Bellekten ID Okuma Mesajları ve Message Type Bilgisi\)](#)
DATA (0x11 0x22 0x33 0x44) → Okunan ID değeri
CRC_HIGH (0xE0) & CRC_LOW (0x7F) → Hata kontrol mekanizmasıdır.

Not:

- MESSAGE_TYPE bilgisi, okunan ID numarasına bağlıdır.
0x50 + ID_INDEX şeklinde hesaplanır.
Örneğin: ID_INDEX = 1 → MESSAGE_TYPE = 0x50 + 0x01 = 0x51
Her ID okuma işleminde, cihaz ilgili ID numarasına göre bir yanıt döndürür.
CRC değerleri, mesajın doğruluğunu sağlamak için kullanılır.

2.Örnek: ID 9 Okuma İşlemi

Bu işlem, ID index 9'daki değeri okumak için kullanılır.

Gönderilen Mesaj Formatı

Command Sent (Command to Read ID 9 in RFID System)							
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	ID_LENGTH	ID_INDEX	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0C	0x02	0x04	0x09	0xBF	0xD4

Alan Açıklamaları

- HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.
DEVICE_ADDRESS (0x01) → Cihazın adresini belirtir.
COMMAND_TYPE (0x0C) → ID işlemlerini belirten ana komuttur.
SUB_COMMAND_TYPE (0x02) → ID okuma işlemini belirtir.
ID_LENGTH (0x04) → Okunacak ID'nin uzunluğunu (4 byte) gösterir.
ID_INDEX (0x09) → Okunacak ID'nin indeksini belirtir.
CRC_HIGH (0xBF) & CRC_LOW (0xD4) → Hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response (ID Read Response for ID 9)					
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	DATA (Read ID 9)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x59	0x6F 0x6E 0x75 0x72	0x9B	0x4E

Alan Açıklamaları

- HEADER (0xAA) → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.
DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresini gösterir.
MESSAGE_TYPE (0x59) → ID okuma işlemini gösterir (MESSAGE_TYPE = 0x50 + ID_INDEX).
DATA (0x6F 0x6E 0x75 0x72) → ID 9'un kaydedilmiş değeri.
CRC_HIGH (0x9B) & CRC_LOW (0x4E) → Hata kontrol mekanizmasıdır.

Not:

- MESSAGE_TYPE bilgisi, okunan ID numarasına bağlıdır.
0x50 + ID_INDEX şeklinde hesaplanır.
Örneğin: ID_INDEX = 9 → MESSAGE_TYPE = 0x50 + 0x09 = 0x59
CRC değerleri, mesajın doğruluğunu sağlamak için kullanılır.

7.5.3.7 ID Sıfırlama Komutu Örneği

Bu örnek, RFID cihazında belirli bir ID değerini sıfırlamak için kullanılan komutun yapısını ve işlem sürecini göstermektedir.

Bu işlem, belirtilen ID indeksindeki ID bilgisini silmek ve varsayılan değere sıfırlamak için kullanılır.

1.Örnek: ID 1 Sıfırlama İşlemi

Bu işlem, ID index 1'deki değeri sıfırlamak için kullanılır.

Gönderilen Mesaj Formatı

Command Sent (Command to Reset ID 1 in RFID System)							
HEADER	DEVICE_ADDRESS	COMMAND_TYPE	SUB_COMMAND_TYPE	ID_LENGTH	ID_INDEX	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0C	0x03	0x04	0x01	0x09	0xEC

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Cihazın adresini belirtir.

COMMAND_TYPE (0x0C) → ID işlemlerini belirten ana komuttur.

SUB_COMMAND_TYPE (0x03) → ID sıfırlama işlemini belirtir.

ID_LENGTH (0x04) → Sıfırlanacak ID'nin uzunluğunu (4 byte) gösterir.

ID_INDEX (0x01) → Sıfırlanacak ID'nin indeksini belirtir.

CRC_HIGH (0x09) & CRC_LOW (0xEC) → Hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response (ID Reset Response for ID 1)				
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x8D	0xC5	0xD5

Alan Açıklamaları:

HEADER (0xAA) → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresini gösterir.

MESSAGE_TYPE (0x8D) → ID sıfırlama işlemine ait mesaj kodudur.

MESSAGE_TYPE değeri, ilgili ID'ye atanmış mesaj kodunun (0x8C) üzerine 1 eklenerek oluşturulmaktadır.

Örneğin, ID okuma mesaj kodu 0x8C iken, sıfırlama işlemi için $0x8C + 1 = 0x8D$ kullanılmaktadır.

CRC_HIGH (0xC5) & CRC_LOW (0xD5) (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

2.Örnek: ID 9 Sıfırlama İşlemi

Gönderilen Mesaj Formatı

Command Sent (Command to Reset ID 9 in the RFID System)							
HEADER	DEVICE _ADDRESS	COMMAND _TYPE	SUB _COMMAND_TYPE	ID _LENGTH	ID _INDEX	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0C	0x03	0x04	0x09	0x88	0xE4

Alan Açıklamaları:

- HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.
- DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın adresidir.
- COMMAND_TYPE (0x0C) → ID işlemleri komutunu belirtir.
- SUB_COMMAND_TYPE (0x03) → ID sıfırlama işlemini belirtir.
- ID_LENGTH (0x04) → Sıfırlanacak ID'nin uzunluğunu belirtir.
- ID_INDEX (0x09) → Sıfırlanacak ID'nin indeksini belirtir.
- CRC_HIGH (0x88) & CRC_LOW (0xE4) (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response (ID Reset Response for ID 9)				
HEADER	DEVICE_ADDRESS	MESSAGE_TYPE	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x95	0x56	0xEC

Alan Açıklamaları:

- HEADER (0xAA) → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.
- DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresini gösterir.
- MESSAGE_TYPE (0x95) → ID sıfırlama işlemine ait mesaj kodudur.
- MESSAGE_TYPE değeri, ilgili ID'ye atanmış mesaj kodunun (0x94) üzerine 1 eklenerek oluşturulmaktadır.
- Örneğin, ID okuma mesaj kodu 0x94 iken, sıfırlama işlemi için $0x94 + 1 = 0x95$ kullanılmaktadır.
- CRC_HIGH (0x56) & CRC_LOW (0xEC) (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

7.5.3.7 Tüm ID'leri Listeleme

Bu komut, RFID cihazında kayıtlı olan tüm ID'leri listelemek için kullanılır.

Bu işlem, cihazın hafızasında bulunan tüm ID değerlerini sorgular ve yanıt olarak döndürür.

Gönderilen Mesaj Formatı

Command Sent (Command to List All IDs in the RFID System)							
HEADER	DEVICE _ADDRESS	COMMAND _TYPE	SUB _COMMAND_TYPE	ID _LENGTH	ID _INDEX	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0C	0x04	0x04	0x00	0x9C	0x5D

Alan Açıklamaları:

HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x0C) → ID işlemleri komutunu belirtir.

SUB_COMMAND_TYPE (0x04) → Tüm ID'leri listeleme işlemini belirtir.

ID_LENGTH (0x04) → Listelenecek ID'lerin uzunluğunu belirtir.

ID_INDEX (0x00) → Başlangıç indeksini belirtir.

CRC_HIGH (0x9C) & CRC_LOW (0x5D) (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response (List All IDs Response)					
HEADER	DEVICE _ADDRESS	MESSAGE _TYPE	DATA (All ID Values)	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0xC8	0x12 0x34 0x56 0x78 0x11 0x22 0x33 0x44 0xAA 0x22 0xCC 0xDD 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xE1 0x2E 0x3D 0x4B 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0xFF 0x6F 0x6E 0x75 0x72 0x5A 0x00	-	-

Alan Açıklamaları:

HEADER (0xAA) → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresini gösterir.

MESSAGE_TYPE (0xC8) → Tüm ID'leri listeleme yanıtı mesaj kodudur.

DATA → Cihazdan okunan tüm ID değerlerini içerir.

0x12 0x34 0x56 0x78 → ID 0

0x11 0x22 0x33 0x44 → ID 1

0xAA 0x22 0xCC 0xDD → ID 2

0xFF 0xFF 0xFF 0xFF → Boş ID (Kayıtlı değil)

0xE1 0x2E 0x3D 0x4B → ID 3

0xFF 0xFF 0xFF 0xFF → Boş ID (Kayıtlı değil)

0x6F 0x6E 0x75 0x72 → ID 9

0x5A 0x00 → Veri sonlandırıcı.

Bu İşlemden Gerçekleşen Adımlar

1. Kullanıcı, tüm ID'leri listelemek için komutu cihaza gönderir.
2. Cihaz, belleğinde bulunan tüm ID değerlerini okur.
3. Mevcut ID bilgileri, yanıt mesajında döndürülür.
4. CRC değerleri, mesajın bütünlüğünü sağlamak için eklenir.

Bu komut, cihazın belleğinde kayıtlı olan tüm ID'leri sorgulamak için kullanılır.

Kayıtlı ID değerleri döndürülür, boş ID alanları ise 0xFF ile işaretlenir.

7.6 Cihaz Yapılandırma Komutları

RFIDAX cihazlarında **Cihaz Yapılandırma Komutları**, cihazın temel çalışma parametrelerini değiştirmek ve yapılandırmak için kullanılır.

Bu komutlar, cihazın ID'sini değiştirme, kullanılan protokolü ayarlama ve haberleşme baud hızını belirleme gibi işlemleri kapsar.

Cihazın uzun vadeli konfigürasyonunu yönetmek ve güncellemek için kullanılan bu komutlar, **Flash Bellek** üzerinde kalıcı değişiklikler yapar.

Bu komutlar yalnızca yetkili kullanıcılar tarafından kullanılabilir ve cihazın çalışmasını doğrudan etkiler.

7.6.1 Cihaz Yapılandırma Komutlarının Kullanım Amaçları

Cihaz yapılandırma komutları, aşağıdaki işlemleri gerçekleştirmek için kullanılır:

1. **Device ID Değiştirme**

Cihazın benzersiz kimlik numarasını (Device ID) değiştirme.

Çoklu cihaz sistemlerinde, her cihazın farklı bir ID'ye sahip olmasını sağlama.

Varsayılan cihaz adresini değiştirme ve Flash Bellek üzerinde güncelleme.

2. **Protokol Değiştirme**

Cihazın haberleşme protokolünü değiştirme.

Kullanılacak haberleşme CRC standardını belirleme (örneğin: CRC16, MODBUS vb.).

Yeni protokol bilgisinin Flash Bellek'e kaydedilmesi.

3. **Baud Rate Değiştirme**

Cihazın haberleşme hızını (baud rate) değiştirme.

Haberleşme hızını belirlenen bir baud rate listesi üzerinden seçme.

Flash Bellek'te kaydedilen baud rate bilgisinin değiştirilmesi.

Bu yapılandırma işlemleri, cihazın farklı sistemlerle uyumlu hale gelmesini ve kullanıcı gereksinimlerine göre özelleştirilmesini sağlar.

7.6.2 Cihaz Yapılandırma Komutları Listesi

Aşağıda **Cihaz Yapılandırma Komutları** ve bunların işlevleri listelenmiştir:

Komut Kodu	Alt Komut Kodu	İşlem
0x0E	0x01	Device ID Değiştirme
0x0E	0x02	Protokol CRC Değiştirme
0x0E	0x03	Baud Rate Değiştirme

Tablo 40 Cihaz Yapılandırma Komutları Listesi



Advanced RFID Card Reader Writer

7.6.3 Device ID Değişirme Komutu

Device ID Değişirme Komutu, cihazın benzersiz adresini değiştirmek için kullanılır.

RFIDAX cihazları, haberleşme sırasında cihaz adresini kullanarak iletişim kurar.

Bu komut sayesinde, aynı sistem içinde birden fazla cihazın birbirinden ayırt edilmesini sağlamak mümkündür.

Device ID değişikliği **Flash Bellek** üzerinde saklanır ve cihaz yeniden başlatıldığında güncellenmiş adresle çalışmaya devam eder.

Not: Maksimum device adresi 255'dir. 255 üstü device adres verilemez.

7.6.3.1 Kullanım Amaçları

Bu komut aşağıdaki amaçlar doğrultusunda kullanılır:

Cihazın haberleşme adresini değiştirmek.

Birden fazla RFID cihazının bulunduğu sistemlerde, her cihaza benzersiz bir adres atamak.

Varsayılan **Device ID** değerini güncellemek.

Flash Bellek üzerine kayıt yaparak **kalıcı adres değişikliği** sağlamak.

7.6.3.2 Gönderilen Mesaj Formatı

Command Sent (Command to Change Device ID in the RFID System)						
HEADER	DEVICE _ADDRESS	COMMAND _TYPE	SUB_ COMMAND_TYPE	NEW _DEVICE_ID	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0E	0x01	0xFF	0xFF	0xFF

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın mevcut adresidir.

COMMAND_TYPE (0x0E) → Cihaz yapılandırma işlemlerini belirten ana komuttur.

SUB_COMMAND_TYPE (0x01) → Device ID değiştirme işlemini belirtir.

NEW_DEVICE_ID (0xFF) → Yeni atanacak **Device ID** değeridir.

CRC_HIGH (XX) & CRC_LOW (XX) (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

7.6.3.3 Gelen Mesaj Formatı (Device ID Değişirme Yanıtı)

Başarılı bir Device ID değiştirme işlemi sonucunda cihaz aşağıdaki yanıtı döndürür:

HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0xFF	0x00	0x00	0xFF	0xFF

Alan Açıklamaları

HEADER (0xAA) → Yanıt mesajının başlangıcını belirler.

DEVICE_ADDRESS (0xFF) → Güncellenmiş **Device ID** adresini gösterir.

STATUS_CODE_HIGH (0x00) & STATUS_CODE_LOW (0x00) → **0x00 0x00** değeri, işlemin başarılı olduğunu gösterir.

CRC_HIGH (XX) & CRC_LOW (XX) (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.



Advanced RFID Card Reader Writer

1. Örnek: Device ID 3'e Güncelleme

Bu örnek, cihazın **Device ID** adresini **0x03** olarak değiştirmek için gönderilen komutu ve alınan yanıtı göstermektedir.

Gönderilen Mesaj (Device ID 3 olarak güncelleme komutu)

Sent Message (Update command to Device ID 3)						
HEADER	DEVICE_ ADDRESS	COMMAND _TYPE	SUB _COMMAND_TYPE	NEW_DEVICE _ID	CRC_HIGH	CRC_LOW
0xAA	0x01	0x0E	0x01	0x03	0x13	0x21

Gelen Mesaj (Başarıyla Güncellendi, Yeni Adres 0x03)

Response Message (Successfully Updated, New Address 0x03)						
HEADER	DEVICE_ ADDRESS (New)	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH	CRC_LOW	
0xBB	0x03	0x00	0x00	0x32	0x5E	

Alan Açıklamaları

HEADER (0xAA - Giden / 0xBB - Gelen) → Komutun başlangıcını ve yanıtın başlangıcını belirler.

DEVICE_ ADDRESS (0x01 - Giden / 0x03 - Gelen) →

Giden mesajda: Komut, mevcut cihaz adresine (0x01) gönderilir.

Gelen mesajda: Cihaz, yeni adresi (0x03) ile yanıt verir.

COMMAND_ TYPE (0x0E) → Cihaz yapılandırma işlemlerini belirten ana komuttur.

SUB_ COMMAND_ TYPE (0x01) → Device ID değiştirme işlemini belirtir.

NEW_DEVICE_ ID (0x03) → Yeni atanacak **Device ID** değeridir.

STATUS_CODE_HIGH (0x00) & STATUS_CODE_LOW (0x00) → **0x00 0x00** değeri, işlemin başarılı olduğunu gösterir.

CRC_HIGH (0x32) & CRC_LOW (0x5E) → Mesajın bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Bu İşlemden Gerçekleşen Adımlar

1. Kullanıcı, cihazın Device ID adresini değiştirmek için komutu gönderir.
2. Cihaz, yeni Device ID değerini Flash Bellek'e kaydeder ve sistemine tanımlar.
3. Yanıt mesajında, yeni güncellenmiş Device ID (0x03) bilgisi ile cevap döndürülür.
4. CRC değerleri, mesajın doğruluğunu sağlamak için eklenir.
5. Cihaz, artık yeni Device ID (0x03) ile haberleşmeye devam eder.

Önemli Notlar

Device ID değişikliği kalıcıdır ve cihaz yeniden başlatıldığında da yeni ID adresiyle çalışmaya devam eder.

Eğer geçersiz bir Device ID değeri gönderilirse, cihaz ERR_INVALID_DEVICE_ADDRESS hatası döndürebilir.

İlgili hata yönetimi için "[6.4.5 İleri Seviye Hatalar ve Sistem Yönetimi](#)" inceleyiniz.

Başarılı değişikliklerden sonra cihazın yeni ID ile haberleşeceği unutulmamalıdır.

Tüm sistemlerde çakışmayı önlemek için her cihazın benzersiz bir ID'ye sahip olması önerilir.

7.6.4 Protokol Değiştirme Komutu

Bu komut, cihazın iletişim protokolünü değiştirmek için kullanılır.

RFIDAX cihazlarında belirlenen protokoller arasında geçiş yapmak için bu komut kullanılır.

Cihaz, belirlenen yeni protokolü **Flash Bellek'e** kaydeder ve gelecek tüm haberleşmelerde bu protokolü kullanır.

Cihaz, desteklenmeyen bir protokol değeri aldığı anda hata döndürür.

Seçilen protokol, cihazın tüm haberleşmelerinde kullanılacaktır.

Gönderilen Mesaj Formatı (Protokol Değiştirme Komutu)

Sent Message Format (Protocol Change Command)						
HEADER	DEVICE_ ADDRESS	COMMAND_ TYPE	SUB_COMMAND_ _TYPE	NEW_PROTOCOL	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xAA	0x01	0x0E	0x02	0xXX	0xXX	0xXX

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın mevcut adresidir.

COMMAND_TYPE (0x0E) → Cihaz yapılandırma işlemlerini belirten ana komuttur.

SUB_COMMAND_TYPE (0x02) → Protokol değiştirme işlemini belirtir.

NEW_PROTOCOL (0xXX) → Yeni atanacak protokol değeri. (Desteklenen CRC Protokolleri aşağıdadır)

CRC_HIGH & CRC_LOW (Opsiyonel) → Mesajın bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Desteklenen Protokoller

Protokol Kodu	Protokol Adı
0x00	USB Protocol
0x01	PROFIBUS Protocol
0x02	MODBUS Protocol
0x03	KERMIT CRC16 Protocol
0x04	CCITT CRC16 Protocol
0x05	ISO/IEC 14443-A Protocol

Tablo 41 Desteklenen Protokoller

Detaaylı detaylı CRC protokolleri için "[6.2.2 Veri bütünlüğü Aktif Olduğu Durum](#)" bakınız.

Önemli Not:

Cihaz, yukarıdaki tabloda yer almayan bir protokol kodu aldığı anda "ERR_INVALID_PROTOCOL" hatası döndürebilir. İlgili hata yönetimi için "[6.4.5 İleri Seviye Hatalar ve Sistem Yönetimi](#)" inceleyiniz.

Bağlantının devam etmesi için, yeni protokol uygun şekilde yapılandırılmalıdır.

Gelen Mesaj Formatı (Protokol Değiştirme Yanıtı)

Response Message Format (Protocol Change Response)					
HEADER	DEVICE_ADDRESS	STATUS_CODE_HIGH	STATUS_CODE_LOW	CRC_HIGH (Optional)	CRC_LOW (Optional)
0xBB	0x01	0x00	0x00	0xFF	0xFF

Alan Açıklamaları (Yanıt Mesajı)

HEADER (0xBB) → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresidir.

STATUS_CODE_HIGH (0x00) & STATUS_CODE_LOW (0x00) →

0x00 0x00 → Protokol başarıyla değiştirildi.

Başka bir hata kodu → Protokol değiştirme başarısız oldu. (Detaylar için [6.4.1 Genel Sistem Durumu ve Hataları](#) bölümüne bakınız.)

CRC_HIGH & CRC_LOW (Opsiyonel) → Mesaj bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Bu İşlemden Gerçekleşen Adımlar

1. Kullanıcı, cihazın protokolünü değiştirmek için komutu gönderir.
2. Cihaz, belirtilen protokolün geçerli olup olmadığını kontrol eder.
3. Protokol geçerliyse, Flash Bellek'e kaydedilir ve cihaz bu protokolü kullanmaya başlar.
4. Başarılı işlem sonucunda, cihaz "0x00 0x00" status kodu ile yanıt döndürür.
5. Yanıt mesajında, mesajın doğruluğunu sağlamak için CRC değerleri eklenir.
6. Eğer geçersiz bir protokol değeri gönderilmişse, cihaz hata mesajı döndürür.

Önemli Notlar

Yeni protokol, cihazın tüm haberleşmelerinde kullanılacaktır.

Cihaz yeniden başlatıldığında, kaydedilen yeni protokol kullanılmaya devam eder.

Geçersiz veya desteklenmeyen bir protokol değeri gönderilirse, cihaz "ERR_INVALID_PROTOCOL" hatası döndürebilir. (Detaylar için [6.4.1 Genel Sistem Durumu ve Hataları](#) bölümüne bakınız.)

Protokol değişikliği sonrası cihaz ile haberleşmeye devam etmek için, yeni protokole uygun bir iletişim yöntemi kullanılmalıdır.

Bu komut, farklı haberleşme yöntemleriyle çalışan sistemlerde cihazın esnekliğini artırmak için kullanılır.

1.Örnek Protokol Güncelleme İşlemi

Gönderilen Mesaj Formatı

Command Sent (Command to Update Protocol in RFID System)						
HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	NEW PROTOCOL	CRC HIGH (Optional)	CRC LOW (Optional)
0xAA	0x01	0x0E	0x02	0x02	0x56	0x53

Alan Açıklamaları

HEADER (0xAA): Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Cihazın adresini belirtir.

COMMAND_TYPE (0x0E): Cihaz yapılandırma işlemlerini belirten ana komuttur.

SUB_COMMAND_TYPE (0x02): Protokol değiştirme işlemini belirtir.

NEW_PROTOCOL (0x02): Seçilen yeni protokol (Bu örnekte **MODBUS CRC16** protokolüne geçiş yapılıyor).

CRC_HIGH (0x56) & CRC_LOW (0x53): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response (Protocol Update Response from RFID System)					
HEADER	DEVICE ADDRESS	STATUS CODE HIGH	STATUS CODE LOW	CRC HIGH (Optional)	CRC LOW (Optional)
0xBB	0x01	0x00	0x00	0x00	0x75

Alan Açıklamaları

HEADER (0xBB): Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Yanıtı gönderen cihazın adresidir.

STATUS_CODE_HIGH (0x00) & STATUS_CODE_LOW (0x00):

0x00 0x00 → Protokol başarıyla değiştirildi.

Eğer hata oluşursa farklı hata kodları dönebilir.

CRC_HIGH (0x00) & CRC_LOW (0x75): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.



Advanced RFID Card Reader Writer

7.6.6 Baud Rate Güncelleme İşlemi

Baud rate, cihazın **haberleşme hızını** belirleyen kritik bir parametredir.

Bu komut, cihazın mevcut **baud rate** değerini değiştirmek için kullanılır.

Baud rate değişimi, **haberleşme performansını optimize etmek, farklı sistemlerle uyumluluğu sağlamak veya belirli bir veri iletim hızına ayarlamak** amacıyla gerçekleştirilir.

Baud rate güncellemesi **cihaz yeniden başlatılmalıdır**.

7.6.6.1 Desteklenen Baud Rate Değerleri

Aşağıda, cihazın desteklediği **baud rate** hızları ve bunlara karşılık gelen **index değerleri** verilmiştir:

Baud Rate (bps)	Baud Rate Index (0xXX)
600	0x00
1200	0x01
2400	0x02
4800	0x03
9600	0x04
14400	0x05
19200	0x06
28800	0x07
38400	0x08
56000	0x09
57600	0x0A
115200	0x0B
128000	0x0C
256000	0x0D

Tablo 42 Desteklenen Baud Rate Değerleri

Baud rate seçimi **index değeri kullanılarak** yapılır. Kullanıcı, belirlediği baud rate'in karşılık gelen index değerini komut içerisine yazmalıdır.

7.6.6.2 Gönderilen Mesaj Formatı

Command Sent (Command to Update Baud Rate in RFID System)

Command Sent (Command to Update Baud Rate in RFID System)						
HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	NEW BAUD RATE INDEX	CRC HIGH (Optional)	CRC LOW (Optional)
0xAA	0x01	0x0E	0x03	0xXX	0xXX	0xXX

Alan Açıklamaları

HEADER (0xAA): Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x0E): Cihaz yapılandırma işlemlerini belirten ana komuttur.

SUB_COMMAND_TYPE (0x03): Baud rate değiştirme işlemini belirtir.

NEW_BAUD_RATE_INDEX (0xXX): Yeni seçilecek **baud rate index** değeridir.

(Yukarıdaki tabloda verilen indexlerden biri kullanılmalıdır.)

CRC_HIGH (0xXX) & CRC_LOW (0xXX): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.



Advanced RFID Card Reader Writer

Gelen Mesaj Formatı

Response (Baud Rate Update Response from RFID System)

Response (Baud Rate Update Response from RFID System)					
HEADER	DEVICE ADDRESS	STATUS CODE HIGH	STATUS CODE LOW	CRC HIGH (Optional)	CRC LOW (Optional)
0xBB	0x01	0x00	0x00	0xFF	0xFF

Gelen Mesaj Açıklamaları

HEADER (0xBB): Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Yanıtı gönderen cihazın adresidir.

STATUS_CODE_HIGH (0x00) & STATUS_CODE_LOW (0x00):

0x00 0x00 → Baud rate başarıyla değiştirildi.

Eğer hata oluşursa farklı hata kodları dönebilir.

CRC_HIGH (0xFF) & CRC_LOW (0xFF): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Bu İşlemden Gerçekleşen Adımlar

Kullanıcı, yeni baud rate değerini içeren komutu cihaza gönderir.

Cihaz, seçilen baud rate'in geçerli olup olmadığını kontrol eder.

Baud rate güncellemesi başarıyla tamamlandıktan sonra, cihaz yanıt mesajı döndürür.

Yeni baud rate cihazın belleğinde saklanır ve haberleşme hızı değiştirilir.

Baud rate güncellemesi cihaz yeniden başlatılmalıdır.

Önemli Notlar:

Baud rate güncellemesi cihaz yeniden başlatılmalıdır.

Desteklenmeyen bir baud rate seçildiğinde, cihaz "ERR_INVALID_BAUD_RATE" hatası döndürebilir. Detaylar için [6.4.1](#)

[Genel Sistem Durumu ve Hataları](#) bölümüne bakınız

Baud rate değişikliği haberleşmeyi anlık olarak etkileyebilir! Yeni baud rate'i desteklemeyen bir sistemde, haberleşme kesilebilir.

Eğer cihazla tekrar iletişim kurulamazsa, önceki baud rate değerleri denenmelidir.



Advanced RFID Card Reader Writer

1 Örnek: Baud Rate Güncelleme İşlemi

Bu örnek, RFID cihazının baud rate değerinin değiştirilmesi için kullanılan komutun yapısını ve işlem sürecini göstermektedir.

Baud rate, cihazın haberleşme hızını belirleyen kritik bir parametredir ve doğru yapılandırılması, **veri iletişiminin sağlıklı bir şekilde gerçekleşmesi** için önemlidir.

Gönderilen Mesaj Formatı

Command Sent (Command to Update Baud Rate in RFID System)						
HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	NEW BAUD RATE INDEX	CRC HIGH (Optional)	CRC LOW (Optional)
0xAA	0x01	0x0E	0x03	0x0B	0xF4	0x4B

Alan Açıklamaları

HEADER (0xAA): Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Komutun gönderildiği cihazın mevcut adresidir.

COMMAND_TYPE (0x0E): Cihaz yapılandırma işlemlerini belirten ana komuttur.

SUB_COMMAND_TYPE (0x03): Baud rate değiştirme işlemini belirtir.

NEW_BAUD_RATE_INDEX (0x0B): Seçilen yeni baud rate indeksidir (115200 baud).

CRC_HIGH (0xF4) & CRC_LOW (0x4B): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı

Response (Baud Rate Update Response from RFID System)					
HEADER	DEVICE ADDRESS	STATUS CODE HIGH	STATUS CODE LOW	CRC HIGH (Optional)	CRC LOW (Optional)
0xBB	0x01	0x00	0x00	0x5C	0x3E

Gelen Mesaj Açıklamaları

HEADER (0xBB): Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01): Yanıtı gönderen cihazın adresidir.

STATUS_CODE_HIGH (0x00) & STATUS_CODE_LOW (0x00):

0x00 0x00 → Baud rate başarıyla değiştirildi.

Hata oluşursa, farklı hata kodları dönebilir.

CRC_HIGH (0x5C) & CRC_LOW (0x3E): Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Bu İşlemden Gerçekleşen Adımlar

Kullanıcı, yeni baud rate değerini içeren komutu cihaza gönderir.

Cihaz, yeni baud rate değerinin geçerli olup olmadığını kontrol eder.

Baud rate güncellemesi başarıyla tamamlandıktan sonra, cihaz yanıt mesajı döndürür.

Cihaz yeniden başlatıldıktan sonra

Yeni baud rate değeri cihazın belleğinde saklanır ve sonraki haberleşmeler bu hızda gerçekleşir.

Önemli Notlar:

1- Yeni baud rate atanırken, haberleşmenin devam edebilmesi için bu değere uygun konfigürasyon yapılmalıdır.

2-Eğer yanlış bir baud rate atanırsa, haberleşme sorunları yaşanabilir.

3-Cihaz yeniden başlatıldıktan sonra, baud rate değişmiş olur.

7.7 Fabrika Ayarlarına Dönüş Komutları

RFIDAX cihazlarında **Fabrika Ayarlarına Dönüş Komutları**, tüm ayarları sıfırlamak, kullanıcı verilerini temizlemek ve cihaz belleğini formatlamak için kullanılır.

Bu komutlar, cihazın **kalıcı olarak** fabrika ayarlarına dönmesini sağlar ve kaydedilmiş tüm yapılandırmaları geri alınamaz şekilde siler.

Fabrika ayarlarına dönüş komutları, aşağıdaki işlemleri gerçekleştirir:

Tüm cihaz ayarlarını sıfırlama.

Flash bellekte kayıtlı olan Format ID bilgisini okuma veya değiştirme.

Komut Kodu	Alt Komut Kodu	İşlem
0x0F	0x01	Tüm ayarları fabrika varsayılanlarına sıfırla
0x0F	0x02	Format RFID kartının ID'sini okuma
0x0F	0x03	Format bayrağını set/reset etme
0x0F	0x04	Manuel olarak Format ID'yi Flash belleğe yaz

Tablo 43 Fabrika Ayarlarına Dönüş Komutları Listesi

Önemli Not:

Bu işlemler **geri alınamaz** ve fabrika ayarlarına dönüşten sonra önceki yapılandırmalar kaybolur. İşlemi gerçekleştirmeden önce, mevcut verilerin yedeğini almanız önerilir.

7.7.1 Tüm Ayarları Fabrika Varsayılanlarına Sıfırlama Komutu

Bu komut, RFIDAX cihazındaki tüm yapılandırma ayarlarını fabrika varsayılanlarına döndürmek için kullanılır. Cihazın varsayılan **Device ID, Baud Rate ve Protokol** ayarları sıfırlanarak Flash Bellek üzerine kaydedilir. İşlem tamamlandıktan sonra cihaz **otomatik olarak yeniden başlatılır** ve varsayılan ayarlarla çalışmaya devam eder.

Bu komut aşağıdaki durumlar için kullanılır:

Cihazın tüm yapılandırmalarını fabrika ayarlarına döndürmek.

Yanlış yapılan yapılandırmaları geri almak.

Haberleşme ayarlarını sıfırlayarak standart konfigürasyona dönmek.

Cihazın sistemsel sorunlarını gidermek amacıyla baştan yapılandırmak.

Komut Gönderildi (Cihazı Fabrika Varsayılanlarına Sıfırlama Komutu)

Command Sent (Command to Reset Device to Factory Defaults)					
HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	CRC HIGH (Optional)	CRC LOW (Optional)
0xAA	0x01	0x0F	0x01	0xFF	0xFF

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın mevcut adresidir.

COMMAND_TYPE (0x0F) → Fabrika ayarlarına dönüş komutlarını belirten ana komuttur.

SUB_COMMAND_TYPE (0x01) → Tüm ayarları fabrika varsayılanlarına sıfırlama işlemini belirtir.

CRC_HIGH & CRC_LOW → Mesaj bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı (Fabrika Ayarlarına Sıfırlama Yanıtı)

Yanıt (RFID Sisteminden Fabrika Ayarlarına Sıfırlama Onayı)					
HEADER	DEVICE ADDRESS	STATUS CODE HIGH	STATUS CODE LOW	CRC HIGH (Optional)	CRC LOW (Optional)
0xBB	0x01	0x00	0x00	0xFF	0xFF

Alan Açıklamaları (Yanıt Mesajı)

HEADER (0xBB) → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresidir.

DURUM_KODU_YÜKSEK (0x00) ve DURUM_KODU_DÜŞÜK (0x00)

0x00 0x00 → Fabrika ayarlarına dönüş işlemi başarıyla tamamlandı.

CRC_HIGH & CRC_LOW → Mesaj bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Bu İşlemden Gerçekleşen Adımlar

1-Kullanıcı, cihazı fabrika varsayılanlarına döndürmek için komutu cihaza gönderir.

2-Cihaz, **varsayılan ayarları** aşağıdaki gibi Flash Bellek'e yazar:

- **Aygıt Kimliği** = 0x01
- **Baud Hızı** = 9600(Baud Hızı Endeksi: 0x04)
- **Protokol** = KERMIT CRC16 (0x03)

3-Fabrika ayarları başarılı şekilde yazıldıktan sonra **"SUCCESSFUL"** mesajı döndürülür.

4-Cihaz **otomatik olarak yeniden başlatılır** ve varsayılan ayarlarla çalışmaya devam eder.

Önemli Notlar

Bu işlem, cihazın tüm ayarlarını sıfırlar ve geri alınamaz!

Cihazın haberleşme bilgileri de sıfırlandığından, yeni ayarlara uygun yapılandırma yapılmalıdır.

Cihaz reset sonrası, 9600 Baud Rate ve KERMIT CRC16 protokolü ile haberleşmeye başlar.

Hatalı bir komut gönderilirse, cihaz **"ERR_FLASH_WRITE_FAILURE"** hatası döndürebilir. Bu hata detayları için ["6.4.5 İleri Seviye Hatalar ve Sistem Yönetimi"](#) bakınız.



Advanced RFID Card Reader Writer

1.Örnek: Fabrika Ayarlarına Sıfırlama İşlemi

Bu örnek, cihazın tüm ayarlarını fabrika varsayılanlarına sıfırlamak için gönderilen komutu ve alınan yanıtı göstermektedir.

Gönderilen Mesaj (Fabrika Ayarlarına Sıfırlama Komutu)

Command Sent (Command to Reset Device to Factory Defaults)					
HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	CRC HIGH	CRC LOW
0xAA	0x01	0x0F	0x01	0x31	0x32

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın mevcut adresidir.

COMMAND_TYPE (0x0F) → Fabrika ayarlarına dönüş komutlarını belirten ana komuttur.

SUB_COMMAND_TYPE (0x01) → Tüm ayarları fabrika varsayılanlarına sıfırlama işlemini belirtir.

CRC_HIGH (0x31) & CRC_LOW (0x32) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj (Fabrika Ayarlarına Sıfırlama Yanıtı)

Response (Factory Reset Confirmation from RFID System)					
HEADER	DEVICE ADDRESS	STATUS CODE HIGH	STATUS CODE LOW	CRC HIGH	CRC LOW
0xBB	0x01	0x00	0x00	0x47	0x89

Alan Açıklamaları (Yanıt Mesajı)

HEADER (0xBB) → Yanıt mesajının başlangıcını belirler.

DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresidir.

DURUM_KODU_YÜKSEK (0x00) ve DURUM_KODU_DÜŞÜK (0x00)

0x00 0x00 → Fabrika ayarlarına dönüş işlemi başarıyla tamamlandı.

CRC_HIGH (0x47) & CRC_LOW (0x89) → Mesaj bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

7.7.2 Format RFID Kartının ID'sini Okuma Komutu

Bu komut, **RFID kartının format ID bilgisini okumak** için kullanılır.

Cihaz, belleğinde saklı olan **format ID** değerini okur ve yanıt olarak döndürür.

Kullanım Amaçları

Bu komut aşağıdaki amaçlarla kullanılmaktadır:

RFID kartlarının format ID değerini okumak

Mevcut format ID bilgisinin doğrulanması

Format işlemlerinin geçerliliğini kontrol etmek

Gönderilen Mesaj Formatı

Command Sent (Command to Read Format RFID Card ID in RFID System)					
HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	CRC HIGH (Optional)	CRC LOW (Optional)
0xAA	0x01	0x0F	0x02	0xFF	0xFF

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın mevcut adresidir.

COMMAND_TYPE (0x0F) → Fabrika ayarlarına dönüş komutlarını belirten ana komuttur.

SUB_COMMAND_TYPE (0x02) → RFID kart format ID'sini okuma işlemini belirtir.

CRC_HIGH (XX) & CRC_LOW (XX) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı (Format RFID Kart ID Yanıtı)

Response (Format RFID Card ID Read Response from RFID System)					
BAŞLIK	CİHAZ ADRESİ	MESAJ TÜRÜ	VERİ (RFID Kimliği Biçimi)	CRC YÜKSEK	CRC DÜŞÜK
0xAA	0x01	0x60	0xFF 0xFF 0xFF 0xFF	0xFF	0xFF

Alan Açıklamaları (Yanıt Mesajı)

HEADER (0xAA) → Yanıt mesajının başlangıcını belirler.

DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresidir.

MESSAGE_TYPE (0x60) → Format RFID ID okuma işlemine ait mesaj kodudur.

Mesaj Type bilgisi için lütfen "[6.7.2.1.5 Format ID Okuma Mesajı ve Message Type Bilgisi](#)" bakınız.

DATA (0xFF 0xFF 0xFF 0xFF) → Format RFID ID değeri.

CRC_HIGH (0xFF) & CRC_LOW (0xFF) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Bu İşlemden Gerçekleşen Adımlar

1-Kullanıcı, **format RFID ID'sini okumak** için komutu gönderir.

2-Cihaz, **Flash Bellek'te** kayıtlı olan **format ID** değerini okur.

3-Okunan ID bilgisi, **yanıt mesajı** olarak geri döndürülür.

4-CRC değerleri, **mesajın bütünlüğünü sağlamak** için eklenir.

1. Örnek: Format RFID Kartının ID'sini Okuma

Bu örnek, RFID kartının format ID bilgisinin okunması için kullanılan komutu ve cihazdan alınan yanıtı göstermektedir.

Gönderilen Mesaj (Format RFID ID Okuma Komutu)

Sent Message (Command to Read Format RFID ID in RFID System)					
HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	CRC HIGH	CRC LOW
0xAA	0x01	0x0F	0x02	0x01	0x51

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın mevcut adresidir.

COMMAND_TYPE (0x0F) → Fabrika ayarlarına dönüş işlemlerini belirten ana komuttur.

SUB_COMMAND_TYPE (0x02) → Format RFID ID'sini okuma işlemini belirtir.

CRC_HIGH (0x01) & CRC_LOW (0x51) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj (Format RFID ID Okuma Yanıtı)

Response (Format RFID ID Read Response from RFID System)					
HEADER	DEVICE ADDRESS	MESSAGE TYPE	DATA (Format RFID ID)	CRC HIGH	CRC LOW
0xAA	0x01	0x60	0x93 0x32 0xEF 0xF6	0xE2	0x20

Alan Açıklamaları

HEADER (0xAA) → Yanıt mesajının başlangıcını belirler.

DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresidir.

MESSAGE_TYPE (0x60) → Format RFID ID okuma işlemine ait mesaj kodudur.

DATA (0x93 0x32 0xEF 0xF6 0xE2 0x20) → Format RFID ID değeri.

CRC_HIGH (0xE2) & CRC_LOW (0x20) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

7.7.3 Format Bayrağını Set/Reset Etme Komutu

Bu komut, **format bayrağını set (aktif) veya reset (pasif) etme** işlemleri için kullanılır.

Format bayrağı, cihazın **format işleminin etkin olup olmadığını belirleyen bir kontrol mekanizmasıdır.**

Kullanım Amaçları

Format ID kartı olup olmadığını bir bayrak ile belirlemek için kullanılır.

Format bayrağının değerine göre sistemin nasıl davranacağını kontrol eder.

Bu bayrak pasif olduğunda cihaz kullanıcıdan Bir Format TAG belirlemesini ister.

Detaylar için lütfen bakınız.

Bu bayrak aktif olduğunda ise cihaz normal çalışmasına devam eder.

Gönderilen Mesaj Formatı (Format Bayrağı Set/Reset Etme Komutu)

Message Sent (Command to Set/Reset Format Flag in RFID System)						
HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	Format Flag	CRC HIGH (Optional)	CRC LOW (Optional)
0xAA	0x01	0x0F	0x03	0xFF	0xFF	0xFF

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın mevcut adresidir.

COMMAND_TYPE (0x0F) → Fabrika ayarlarına dönüş işlemlerini belirten ana komuttur.

SUB_COMMAND_TYPE (0x03) → Format bayrağını set/reset etme işlemini belirtir.

FORMAT_FLAG (0xFF) → Bayrak değerini belirler:

0x00 → Format bayrağını reset eder (Flash belleği temizler, format işlemi iptal edilir).

0x01 → Format bayrağını set eder (Flash belleğe 0x00000001 yazılır, format işlemi aktif hale getirilir).

CRC_HIGH & CRC_LOW (Opsiyonel) → Mesajın bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Bu işlem sonrası cihaz bir geridönüş sağlamaz belirli tepkiler verir.

Bu İşlemden Gerçekleşen Adımlar

Set Etme (Aktif Hale Getirme - 0x01 Gönderildiğinde)

- 1-Kullanıcı, **format işlemini etkinleştirmek için** komutu cihaza gönderir.
- 2-Cihaz, **Flash Bellek'te ilgili format bayrağını 0x00000001 olarak günceller.**
- 3-Format bayrağı set edilir ve cihaz format tag istemez.
- 4- Format tag ID'si FF FF FF FF olur.
- 5- Cihaz normal çalışmaya devam eder.

Reset Etme (Pasif Hale Getirme - 0x00 Gönderildiğinde)

- 1-Kullanıcı, **format tag bayrağını iptal etmek için** komutu cihaza gönderir.
- 2-Cihaz, **Flash Bellek'te kayıtlı olan format bayrağını 0xFFFFFFFF olarak günceller.**
- 3-Format cihaz bir format tag ister.
- 4-Yanıt mesajında, **durmadan format tag talep eder.**

Format tag okuna denk "ERR_RESET_CARD_NOT_READ" hatasını verir

Lütfen "[6.4.5 İleri Seviye Hatalar ve Sistem Yönetimi](#)" bakınız.

Bu hatadan kurtulma yolu ise ya format tag tanımlamaktır ya da format tag bayrağını 0x01 yapmaktır.

7.7.3.1 Format Bayrağını Reset Etme (0x00 - Pasif Hale Getirme)

Bu, cihazın format bayrağını sıfırlamak (reset etmek) için gönderilen komutu ve cihazdan gelen yanıtı gösterir.

Gönderilen Mesaj (Format Bayrağını Reset Etme Komutu)

Command Sent (Command to Reset Format Flag in RFID System)						
HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	Format Flag	CRC HIGH (Optional)	CRC LOW (Optional)
0xAA	0x01	0x0F	0x03	0x00	0x72	0x10

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x0F) → Fabrika ayarlarına dönüş işlemlerini belirten ana komuttur.

SUB_COMMAND_TYPE (0x03) → Format bayrağını set/reset etme işlemini belirtir.

FORMAT_FLAG (0x00) → Format bayrağını sıfırlar (Flash belleği temizler, format işlemi iptal edilir).

CRC_HIGH (0x72) & CRC_LOW (0x10) → Mesajın bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Yanıt Mesajı (Hata Kodu İçeren Yanıt)

Incoming Response Message (Response Containing Error Code)						
HEADER	DEVICE ADDRESS	STATUS CODE HIGH	STATUS CODE LOW	CRC HIGH	CRC LOW	ERROR CODE DESCRIPTION
0xBB	0x01	0x00	0x71	0x32	0x88	ERR_RESET_CARD_NOT_READ

Alan Açıklamaları

HEADER (0xBB) → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresidir.

DURUM_KODU_YÜKSEK (0x00) ve DURUM_KODU_DÜŞÜK (0x71) →

0x00 0x71, cihazın format bayrağını reset ettiği ancak format tag'ın okunmadığını gösteren hata kodudur (ERR_RESET_CARD_NOT_READ).

CRC_HIGH (0x32) & CRC_LOW (0x88) → Mesajın bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Hata Kodu Açıklaması (ERR_RESET_CARD_NOT_READ)

Bu hata, cihazın format bayrağını reset ettikten sonra format tag okunmadığı anlamına gelir.

Cihaz, bir format tag olmadan normal çalışmaya devam edemez.

Sistem, sürekli format tag arayarak "ERR_RESET_CARD_NOT_READ" hatası döndürmeye devam edecektir.

Lütfen "[6.4.5 İleri Seviye Hatalar ve Sistem Yönetimi](#)" bakınız.

Çözüm Yöntemleri

Bu hatadan kurtulmak için aşağıdaki iki çözümün biri uygulanmalıdır:

1-Format Tag Okutulmalı:

Cihazın beklediği format tag bir RFID okuyucuya okutulmalı ve tanımlanmalıdır.

Format tag doğru şekilde okutulduğunda, cihaz normal çalışmasına dönecektir.

2-Format Bayrağı Tekrar Set Edilmeli:

Format bayrağı tekrar "0x01" yapılarak cihazın format modundan çıkması sağlanabilir.

Bunun için Format Bayrağını Set Etme (0x01) komutu gönderilmelidir.

7.7.3.2 Format Bayrağını Set Etme (0x01 - Aktif Hale Getirme)

Bu örnek, cihazın **format bayrağını aktif hale getirmek (set etmek)** için gönderilen komutu ve cihazın verdiği tepkiyi gösterir.

Gönderilen Mesaj (Format Bayrağını Set Etme Komutu)

Komut Gönderildi (RFID Sisteminde Biçim Bayrağı Ayarlama Komutu)

Command Sent (Command to Set Format Flag in RFID System)						
HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	FORMAT FLAG	CRC HIGH	CRC LOW
0xAA	0x01	0x0F	0x03	0x01	0x62	0x31

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın adresidir.

COMMAND_TYPE (0x0F) → Fabrika ayarlarına dönüş işlemlerini belirten ana komuttur.

SUB_COMMAND_TYPE (0x03) → Format bayrağını set/reset etme işlemini belirtir.

FORMAT_FLAG (0x01) → Format bayrağını aktif hale getirir (Flash belleğe **0x00000001** yazılır).

CRC_HIGH (0x62) & CRC_LOW (0x31) → Mesajın bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Cihazın Tepkisi

Bu komuta karşılık cihaz, bir yanıt mesajı döndürmez.

Cihaz, Flash Bellek'te ilgili format bayrağını "**0x00000001**" olarak günceller.

Format bayrağı set edilir ve cihaz format tag istemez.

Format tag ID'si "**FF FF FF FF**" olur.

Cihaz normal çalışmaya devam eder.

Bu İşlemden Gerçekleşen Adımlar

Set Etme (Aktif Hale Getirme - 0x01 Gönderildiğinde)

1-Kullanıcı, **format işlemini etkinleştirmek için** komutu cihaza gönderir.

2-Cihaz, **Flash Bellek'te ilgili format bayrağını 0x00000001 olarak günceller.**

3- **Format bayrağı set edilir ve cihaz format tag istemez.**

4-**Format tag ID'si FF FF FF FF olur.**

5-**Cihaz normal çalışmaya devam eder.**

Önemli Notlar

Bu komut, cihazın format bayrağını değiştirir ancak cihazın tamamen formatlanmasını sağlamaz.

Format bayrağı sıfırlandığında cihaz bir format tag okumadığı sürece sürekli hata döndürebilir.

Format bayrağı set edildiğinde cihaz normal çalışmaya devam eder ve bir format tag aramaz.

Eğer yanlışlıkla format bayrağı sıfırlanırsa, cihazı tekrar çalıştırmak için bir format tag okutulmalı veya bayrak tekrar "**0x01**" yapılmalıdır.

İşlem	Sonuç	Hata Kodu
Format Bayrağını Reset Etme (0x00)	Cihaz format tag beklemeye başlar ve sürekli hata döndürür.	HATA_KART_SIFIRLA_OKUNMADI (0x00 0x71)
Format Bayrağını Set Etme (0x01)	Cihaz normal çalışmaya devam eder, format tag istemez.	Cihaz herhangi bir yanıt döndürmez.

Tablo 44 Format Bayrağını Set (aktif) veya Reset (pasif) Durumu

Sonuç

Cihazın format bayrağını sıfırlamak hata mesajına neden olur ve cihaz bir format tag bekler.

Bu hatadan kurtulmak için ya bir format tag okutulmalı ya da bayrak tekrar "**0x01**" yapılmalıdır

7.7.4 Manuel Olarak Format ID'yi Flash Belleğe Yazma

Bu komut, kullanıcı tarafından manuel olarak belirlenen bir Format ID değerini doğrudan Flash belleğe yazmak için kullanılır.

Cihazın format mekanizmasında kullanılacak olan bu ID, sistem tarafından belirli işlemler için tanımlanır ve Flash bellekte saklanır.

Kullanım Amaçları

Kullanıcı tarafından belirlenen bir Format ID'yi cihazın Flash belleğine manuel olarak yazmak.

Format işlemi sırasında, cihazın önceden belirlenen bir kimlik doğrulama mekanizmasına sahip olmasını sağlamak.

Belirlenen Format ID'nin sistem tarafından kabul edilip edilmediğini doğrulamak.

Gönderilen Mesaj Formatı

Sent Message Format						
HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	FORMAT ID (4 Bytes)	CRC HIGH (Optional)	CRC LOW (Optional)
0xAA	0x01	0x0F	0x04	0xFF 0xFF 0xFF 0xFF	0xFF	0xFF

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın mevcut adresidir.

COMMAND_TYPE (0x0F) → Fabrika ayarlarına dönüş işlemlerini belirten ana komuttur.

SUB_COMMAND_TYPE (0x04) → Manuel olarak bir Format ID değerini Flash belleğe yazma işlemini belirtir.

FORMAT_ID (0xFF 0xFF 0xFF 0xFF) → Flash belleğe kaydedilecek 4 baytlık Format ID verisi.

CRC_HIGH & CRC_LOW (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj Formatı (Format ID Yazma Yanıtı)

Incoming Message Format (Format ID Write Response)					
HEADER	DEVICE ADDRESS	STATUS CODE HIGH	STATUS CODE LOW	CRC HIGH (Optional)	CRC LOW (Optional)
0xBB	0x01	0x00	0x00	0xFF	0xFF

Gelen Mesaj Açıklamaları

HEADER (0xBB) → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresidir.

STATUS_CODE_HIGH (0x00) & STATUS_CODE_LOW (0x00)

0x00 0x00 → Format ID başarıyla Flash belleğe yazıldı.

Farklı bir hata kodu → Format ID yazma başarısız oldu. (Detaylar için **6.4.1 Genel Sistem Durumu ve Hataları** bölümüne bakınız.)

CRC_HIGH & CRC_LOW (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Bu İşlemden Geçerleşen Adımlar

1-Kullanıcı, belirlenen bir Format ID'yi Flash belleğe kaydetmek için komutu gönderir.

2-Cihaz, gelen Format ID'nin geçerli olup olmadığını kontrol eder.

3-Eğer Format ID geçerli bir HEX formatındaysa, cihaz Flash belleği temizler ve yeni ID değerini yazar.

4-Başarılı işlem sonucunda, cihaz "0x00 0x00" status kodu ile yanıt döndürür.

5-Cihaz, bundan sonraki format işlemlerinde, Flash bellekte kaydedilen yeni Format ID'yi kullanır.

Önemli Notlar

Format ID yalnızca HEX formatında olmalıdır. 0xFF'den büyük bir değer gönderildiğinde **ERR_INVALID_HEX_DATA** hatası dönebilir. Detaylı bilgi için lütfen "**6.4.5 İleri Seviye Hatalar ve Sistem Yönetimi**" bakınız.

Flash bellek silme işlemi geri alınamaz. Yeni Format ID yazıldığında eski değer geri getirilemez.

Geçersiz bir Format ID gönderildiğinde, cihaz hata kodu döndürerek işlemi iptal eder.

Format ID başarıyla kaydedildiğinde, cihaz bundan sonra bu ID'yi kullanarak işlem yapacaktır.



Advanced RFID Card Reader Writer

1.Örnek: Manuel Olarak Format ID'yi Flash Belleğe Yazma İşlemi

Bu örnek, kullanıcının manuel olarak belirlediği **Format ID** değerini Flash belleğe yazmak için gönderilen komutu ve alınan yanıtı göstermektedir.

Gönderilen Mesaj (Format ID Yazma Komutu)

Sent Message (Command to Write Format ID to Flash Memory)						
HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	FORMAT ID (4 Bytes)	CRC HIGH	CRC LOW
0xAA	0x01	0x0F	0x04	0x13 0x75 0x41 0xFC	0x71	0x22

Alan Açıklamaları

HEADER (0xAA) → Komutun başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Komutun gönderildiği cihazın mevcut adresidir.

COMMAND_TYPE (0x0F) → Fabrika ayarlarına dönüş işlemlerini belirten ana komuttur.

SUB_COMMAND_TYPE (0x04) → Manuel olarak bir Format ID değerini Flash belleğe yazma işlemini belirtir.

FORMAT_ID (0x13 0x75 0x41 0xFC) → Flash belleğe kaydedilecek 4 baytlık Format ID verisi.

CRC_HIGH (0x71) & CRC_LOW (0x22) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

Gelen Mesaj (Format ID Yazma Yanıtı)

Response Message (Format ID Write Response)						
HEADER	DEVICE ADDRESS	STATUS CODE HIGH	STATUS CODE LOW	CRC HIGH	CRC LOW	
0xBB	0x01	0x00	0x00	0xFF	0xFF	

Gelen Mesaj Açıklamaları

HEADER (0xBB) → Yanıt mesajının başlangıcını belirler. Sabit bir değerdir.

DEVICE_ADDRESS (0x01) → Yanıtı gönderen cihazın adresidir.

STATUS_CODE_HIGH (0x00) & STATUS_CODE_LOW (0x00)

0x00 0x00 → Format ID başarıyla Flash belleğe yazıldı.

Eğer hata oluşursa farklı hata kodları dönebilir.

CRC_HIGH & CRC_LOW (Opsiyonel) → Veri bütünlüğünü sağlamak için kullanılan hata kontrol mekanizmasıdır.

7.8 Sector Trailer Yönetim Komutları

RFIDAX cihazlarında **Sector Trailer Yönetim Komutları**, RFID kart üzerindeki her sektörün erişim kontrol yapılarını oluşturmak ve düzenlemek için kullanılır.

Her sector trailer yapısı, iki erişim anahtarı (**KeyA** ve **KeyB**) ile erişim izinlerini belirleyen Access Bits alanından oluşur.

Bu komutlar çalıştırılırken cihazın belleğinde daha önceden tanımlanmış ve kaydedilmiş olan opsiyonel KeyA ve KeyB değerleri kullanılır.

Bu anahtarlar, kullanıcı tarafından belirlenmiş özel anahtarlardır ve detayları [7.3.2.5.2 Kullanıcı Tarafından Tanımlanabilen Anahtarlar \(Optional KeyA ve Optional KeyB\)](#) bölümünde açıklanmıştır.

Bu yapı sayesinde, kart üzerindeki sektörlerin erişim güvenliği standart ve güvenli bir formatla yeniden yapılandırılır.

Erişim izinlerini kontrol eden **Access Bits** alanı sabit olarak 7F 07 88 40 değerinde belirlenir.

Komut Kodu	Alt Komut	Açıklama
0x11	0x01	Kartla iletişime geçmeden, cihazın belleğinde kayıtlı opsiyonel KeyA/KeyB ve sabit Access Bits ile sector trailer veri yapısı oluşturulur ve dış sistemlere gönderilir.
0x11	0x02	Kart üzerindeki sadece sector trailer blokları, cihazda kayıtlı opsiyonel KeyA/KeyB anahtarları kullanılarak güncellenir. Veri bloklarına müdahale edilmez.
0x11	0x03	Kartın tüm veri blokları sıfırlanır (0x00 yazılır) ve ardından her trailer bloğu, cihazda kayıtlı opsiyonel KeyA/KeyB ve sabit Access Bits ile yeniden yapılandırılır.
0x11	0x04	Kartın tüm blokları silinir ve her sektör tamamen opsiyonel KeyA/KeyB ve sabit Access Bits kullanılarak formatlanır.

Tablo 45 Sector Trailer Yönetim Komutları Listesi

Önemli Notlar:

- İşlemler sırasında, yalnızca cihaz belleğinde önceden kullanıcı tarafından tanımlanarak kaydedilmiş olan **opsiyonel KeyA ve KeyB anahtarları** kullanılır. Detaylı bilgi için lütfen [7.3.2.5.2 – Kullanıcı Tarafından Tanımlanabilen Anahtarlar \(Optional KeyA ve Optional KeyB\)](#) bölümüne bakınız.
- İşlem sonrası kartın erişim kontrol yapısı değişir ve eski anahtarlar geçersiz hale gelir.
- Kart üzerindeki veri kaybını önlemek için işlemlerden önce veri yedeği alınması tavsiye edilir.

7.8.1 Sector Trailer Okuma Komutu

Bu komut, cihazın belleğinde saklanan opsiyonel KeyA ve KeyB anahtarları ile sabit Access Bits (7F 07 88 40) değerini kullanarak standart bir Sector Trailer veri yapısı oluşturur.

Kart ile fiziksel bir iletişim kurulmaz.

Oluşturulan veri, dış sistemlere gönderilir ve kart formatlama veya güvenlik yapılandırılmalarında referans veri olarak kullanılabilir.

Kullanım Amaçları

- Cihaz belleğinde tanımlı opsiyonel KeyA ve KeyB anahtarları ile standart bir sector trailer veri paketi oluşturmak.
- Formatlama veya erişim kontrol senaryoları için güvenli referans veri sağlamak.
- Dış sistemlere gönderilmek üzere önceden hazırlanmış erişim yapısını üretmek.

HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB TYPE	COMMAND	CRC (Optional)	HIGH	CRC (Optional)	LOW
0xAA	0x01	0x11	0x01		0xFF		0xFF	

Alan Açıklamaları

- **HEADER (0xAA):** Komutun başlangıcını belirler.
- **DEVICE_ADDRESS (0x01):** Komutun gönderildiği cihazın adresidir.
- **COMMAND_TYPE (0x11):** Sector Trailer işlemlerini temsil eden ana komuttur.
- **SUB_COMMAND_TYPE (0x01):** Sector Trailer veri yapısı oluşturma alt komutudur.
- **CRC_HIGH & CRC_LOW (Opsiyonel):** Veri bütünlüğü için kullanılan hata kontrol alanlarıdır.

Gelen Mesaj Formatı

(Incoming Message Format)

HEADER	DEVICE ADDRESS	MESSAGE TYPE	KeyA (6 Byte)	Access Bits (4 Byte)	KeyB (6 Byte)	CRC HIGH (Optional)	CRC LOW (Optional)
0xAA	0x01	0x13	6 Byte	4 Byte	6 Byte	0xFF	0xFF

Gelen Mesaj Açıklamaları

- **HEADER (0xAA):** Yanıt mesajının başlangıcını belirtir.
- **DEVICE_ADDRESS (0x01):** Yanıtı gönderen cihazın adresidir.
- **MESSAGE_TYPE (0x13):** Gönderilen verinin Sector Trailer veri yapısı olduğunu belirtir.
- **KeyA:** Cihazda kayıtlı opsiyonel KeyA anahtar verisi (6 bayt).
- **Access Bits:** Sabit erişim kontrol verisi (7F 07 88 40 - 4 bayt).
- **KeyB:** Cihazda kayıtlı opsiyonel KeyB anahtar verisi (6 bayt).
- **CRC_HIGH & CRC_LOW (Opsiyonel):** Veri bütünlüğü kontrolü için kullanılan alanlardır.

Bu İşlemden Gerçekleşen Adımlar

1. Kullanıcı, sector trailer veri yapısının oluşturulması için cihazın haberleşme arayüzüne AA 01 11 01 formatında bir HEX komutu iletimini gerçekleştirir.
2. Cihaz, belleğinde kayıtlı opsiyonel KeyA, KeyB ve sabit Access Bits bilgilerini kullanarak 16 baytlık Sector Trailer veri yapısını oluşturur.
3. Cihaz, oluşturduğu veri yapısını 0x13 mesaj tipi ile dış sisteme gönderir.
4. Dış sistem gelen veri ile KeyA, Access Bits ve KeyB yapılarını alır.



Advanced RFID Card Reader Writer

1. Örnek: Sector Trailer Veri Yapısı Oluşturma İşlemi

Bu örnekte, kullanıcı cihazın belleğinde daha önce tanımlanmış opsiyonel KeyA, KeyB anahtarları ve sabit erişim bitleri kullanılarak bir sector trailer veri yapısının oluşturulmasını sağlar. Kart ile fiziksel iletişim kurulmaz; veri doğrudan cihazın iç belleğinden üretilir ve dış sisteme gönderilir.

HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE
0xAA	0x01	0x11	0x01

Gönderilen Komut Açıklamaları:

- HEADER (0xAA):** Komut başlangıcını belirtir.
- DEVICE_ADDRESS (0x01):** Cihazın adresi.
- COMMAND_TYPE (0x11):** Sector Trailer Yönetim Komutları.
- SUB_COMMAND_TYPE (0x01):** Cihaz belleğindeki verilerle sector trailer veri yapısını oluşturur.

Gelen HEX Yanıt

HEADER	DEVICE ADDRESS	MESSAGE TYPE	KeyA (6 Byte)	Access Bits (4 Byte)	KeyB (6 Byte)
0xAA	0x01	0x13	0x1A 0x2A 0x3A 0x4A 0x5A 0x6A	0x7F 0x07 0x88 0x40	0xA1 0xB2 0x99 0xD1 0xE1 0xF1

Gelen Yanıt Açıklamaları

HEADER (0xAA): Gelen veri paketinin başlangıcını belirtir ve sabit değerdedir. Yanıtın bir veri aktarımı olduğunu gösterir.

DEVICE_ADDRESS (0x01): Mesajı gönderen cihazın benzersiz adres bilgisidir. İletişimde hedef cihazı tanımlar.

MESSAGE_TYPE (0x13): Cihazın oluşturduğu sector trailer veri yapısını taşıyan yanıt tipidir. Bu mesaj, KeyA, Access Bits ve KeyB alanlarını içerir.

KeyA (0x1A 0x2A 0x3A 0x4A 0x5A 0x6A): Bellekte kullanıcı tarafından önceden tanımlanmış opsiyonel KeyA anahtar değerlerini temsil eder. Her bir sektör için kimlik doğrulamada kullanılacak 6 baytlık güvenlik anahtarıdır.

Access Bits (0x7F 0x07 0x88 0x40): Kartın veri bloklarına atanan erişim izinlerini tanımlayan sabit 4 baytlık alandır. "7F 07 88 40" kombinasyonu standart güvenlik yapılandırmasını temsil eder.

KeyB (0xA1 0xB2 0x99 0xD1 0xE1 0xF1): Bellekte kullanıcı tarafından tanımlanan opsiyonel KeyB anahtar değerlerini içerir. İkinci kimlik doğrulama ve veri erişim anahtarı olarak kullanılır.

CRC Bilgisi: Bu yanıt mesajında **CRC alanı bulunmamaktadır**. Veri bütünlüğü kontrolü yapılmaz; doğrudan oluşturulan sector trailer veri yapısı dış sisteme iletilir.

Önemli Notlar:

- İşlem sırasında kart ile fiziksel bağlantı kurulmaz, veri doğrudan cihaz belleğinden oluşturulur.
- Bu anahtarlar, kullanıcı tarafından belirlenmiş özel anahtarlardır ve detayları [7.3.2.5.2 Kullanıcı Tarafından Tanımlanabilen Anahtarlar \(Optional KeyA ve Optional KeyB\)](#) bölümünde açıklanmıştır.
- Erişim bitleri sabit olarak **7F 07 88 40** değerinde ayarlanmıştır.
- Komut yalnızca **MODE_ADVANCED** çalıştırılabilir.

7.8.2 Yalnızca Sector Trailer Bloklarını Özel Key'lerle Güncelleme

Bu komut, kart üzerindeki yalnızca **sector trailer bloklarını**, cihaz belleğinde daha önce tanımlanmış opsiyonel **KeyA**, **KeyB** anahtarları ve sabit **Access Bits (7F 07 88 40)** değerini kullanarak günceller.

İşlem sırasında **veri bloklarına müdahale edilmez**, sadece her sektörün trailer yapısı yeniden yapılandırılır.

Kart ile fiziksel bağlantı sağlanır, kimlik doğrulama (authentication) yapılır ve doğrulama başarılı olursa ilgili sector trailer güncellenir.

Kullanım Amaçları

- RFID kartın güvenlik yapılandırmasını yeniden oluşturmak.
- Sadece erişim kontrol yapısını değiştirmek, veri bloklarını korumak.
- Mevcut güvenlik anahtarlarını daha güvenli KeyA/KeyB kombinasyonlarıyla değiştirmek.

Gönderilen Mesaj Formatı

HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	KEY TYPE	AUTH TYPE	CRC HIGH (Optional)	CRC LOW (Optional)
0xAA	0x01	0x11	0x02	0xFF	0xFF	0xFF	0xFF

Alan Açıklamaları

- **HEADER (0xAA):** Komutun başlangıcını tanımlar.
- **DEVICE_ADDRESS (0x01):** Komutun hedeflendiği cihaz adresidir.
- **COMMAND_TYPE (0x11):** Sector Trailer yönetim işlemleri için kullanılan ana komuttur.
- **SUB_COMMAND_TYPE (0x02):** Yalnızca sector trailer bloklarını güncelleme alt komutudur.
- **KEY_TYPE:**
 - 0x01 → KeyA kullan.
 - 0x02 → KeyB kullan.
 - 0x03 → Opsiyonel KeyA kullan (Bkz: 7.3.2.5.2).
 - 0x04 → Opsiyonel KeyB kullan (Bkz: 7.3.2.5.2).
 - (Detaylı bilgi için bkz: [7.3.2.5.2 Kullanıcı Tarafından Tanımlanabilen Anahtarlar \(Optional KeyA ve Optional KeyB\)](#))
- **AUTH_TYPE:**
 - 0x01 → AUTHENT1A ile doğrulama.
 - 0x02 → AUTHENT1B ile doğrulama.
- **CRC_HIGH & CRC_LOW (Optional):** Veri bütünlüğü kontrolü için kullanılan alanlardır.

Bu İşlemden Gerçekleşen Adımlar

1. Kullanıcı, 0xAA 0x01 0x11 0x02 formatında bir HEX komutu gönderir ve KEY_TYPE, AUTH_TYPE bilgilerini belirtir.
2. Cihaz, kart ile fiziksel iletişim kurarak kimlik doğrulaması yapar.
3. Kimlik doğrulama başarılı olursa, kart üzerindeki her sector trailer bloğu yeni KeyA/KeyB ve sabit Access Bits ile güncellenir.
4. Veri bloklarına (data blocks) müdahale edilmeden sadece trailer bloklar değiştirilir.
5. Başarılı güncelleme durumunda cihaz, "0x00 0x00" status kodu ile cevap verir.
6. Eğer cevapta hata kodu dönerse, hata türünü ve detaylarını öğrenmek için [6.4.4 RFID Kart Okuma/Yazma Hataları](#) bölümüne başvurulmalıdır.



Advanced RFID Card Reader Writer

1. Örnek: Yalnızca Sector Trailer Bloklarını Güncelleme İşlemi

Bu örnekte, kullanıcı kart üzerindeki yalnızca **sector trailer** bloklarını, cihazın belleğinde önceden tanımlı opsiyonel **KeyA** veya **KeyB** anahtarlarını kullanarak günceller. Veri bloklarına müdahale edilmez, sadece erişim kontrol bilgileri değiştirilir.

Gönderilen Komut

HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	KEY TYPE	AUTH TYPE	CRC HIGH	CRC LOW
0xAA	0x01	0x11	0x02	0x01	0x02	0xD3	0x6B

Alan Açıklamaları:

- **HEADER (0xAA):** Komut başlangıcı.
- **DEVICE_ADDRESS (0x01):** Cihaza ait adres.
- **COMMAND_TYPE (0x11):** Sector Trailer işlemleri komutu.
- **SUB_COMMAND_TYPE (0x02):** Sadece trailer bloklarını güncelleme alt komutu.
- **KEY_TYPE (0x01):** KeyA anahtarı kullanılacak.
- **AUTH_TYPE (0x02):** PICC_AUTHENT1B ile kimlik doğrulama yapılacaktır.
- **CRC (0xD3 0x6B):** Veri bütünlüğü doğrulaması için eklenmiş opsiyonel hata kontrol kodu.

HEADER	DEVICE ADDRESS	STATUS CODE HIGH	STATUS CODE LOW	CRC HIGH	CRC LOW
0xBB	0x01	0x00	0x00	0x5C	0x3E

Gelen Yanıt Açıklamaları:

- **HEADER (0xBB):** Gelen yanıtın başlangıcı.
- **DEVICE_ADDRESS (0x01):** Yanıtı gönderen cihazın adresi.
- **STATUS_CODE_HIGH (0x00) & STATUS_CODE_LOW (0x00):** İşlemin başarıyla tamamlandığını gösterir.
- **CRC (0x5C 0x3E):** Yanıt veri bütünlüğü kontrol alanı (opsiyonel)

Bu İşlemden Gerçekleşenler

- Kullanıcı, cihazın haberleşme arayüzüne **AA 01 11 02 01 02 D3 6B** komutunu gönderir.
- Cihaz, kartı algılar ve seçilen kimlik doğrulama yöntemi ile sector trailer bloklarına erişir.
- Her sector trailer bloğu, seçilen KeyA ve sabit erişim bitleri kullanılarak güncellenir.
- Veri bloklarına (data blocks) hiçbir müdahalede bulunulmaz.
- Başarılı tamamlanan işlem sonrasında cihaz, **0x00 0x00** status kodu ile yanıt verir.

7.8.3 Tüm Veri Bloklarını Sıfırlama ve Sector Trailer Yapılandırması

Bu komut, kart üzerindeki **bütün veri bloklarını (data blocks)** sıfırlayarak (0x00 yazılarak) temizler ve her sektörün **sector trailer bloklarını** cihazın belleğinde tanımlı opsiyonel **KeyA, KeyB** anahtarları ve sabit **Access Bits (7F 07 88 40)** ile yeniden yapılandırır. İşlem kart ile fiziksel iletişim kurularak yapılır ve tüm bloklar kimlik doğrulama sonrası sıfırlanır veya güncellenir.

Kullanım Amaçları

- Tüm kartı sıfırlayarak temiz bir güvenlik altyapısı oluşturmak.
- Eski verileri ve erişim yapılandırmalarını tamamen kaldırmak.
- Yeni KeyA, KeyB ve sabit erişim bitleriyle kartı yeniden yapılandırmak.

Gönderilen Mesaj Formatı

HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	KEY TYPE	AUTH TYPE	CRC HIGH (Optional)	CRC LOW (Optional)
0xAA	0x01	0x11	0x03	0xFF	0xFF	0xFF	0xFF

Alan Açıklamaları

- **HEADER (0xAA)**: Komutun başlangıcını belirler.
- **DEVICE_ADDRESS (0x01)**: Hedef cihazın adresi.
- **COMMAND_TYPE (0x11)**: Sector Trailer yönetim işlemleri komutu.
- **SUB_COMMAND_TYPE (0x03)**: Tüm blokları sıfırlama ve trailer yapılandırma alt komutu.
- **KEY_TYPE**:
 - 0x01 → KeyA kullanılır.
 - 0x02 → KeyB kullanılır.
 - 0x03 → Opsiyonel KeyA kullanılır
 - 0x04 → Opsiyonel KeyB kullanılır(Detaylı bilgi için bkz: [7.3.2.5.2 Kullanıcı Tarafından Tanımlanabilen Anahtarlar \(Optional KeyA ve Optional KeyB\)](#))
- **AUTH_TYPE**:
 - 0x01 → AUTHENT1A doğrulaması yapılır.
 - 0x02 → AUTHENT1B doğrulaması yapılır.
- **CRC_HIGH & CRC_LOW** (Opsiyonel): Veri bütünlüğü kontrolü için eklenir.

Bu İşlemden Gerçekleşen Adımlar

1. **Kullanıcı**, 0xAA 0x01 0x11 0x03 formatında bir HEX komutu gönderir ve KEY_TYPE ile AUTH_TYPE bilgilerini belirtir.
2. **Cihaz**, kartı algılar ve seçilen kimlik doğrulama yöntemi ile tüm bloklara erişim sağlar.
3. **Her veri bloğu** (data block) 0x00 ile sıfırlanır.
4. **Her sector trailer bloğu**, opsiyonel KeyA/KeyB ve sabit Access Bits (7F 07 88 40) ile yeniden yapılandırılır.
5. **Başarılı** bir işlemde cihaz, "0x00 0x00" status kodu ile yanıt verir.
6. Eğer cevapta hata kodu bulunursa, detaylı bilgi için [6.4.4 RFID Kart Okuma/Yazma Hataları](#) bölümüne başvurulmalıdır.



Advanced RFID Card Reader Writer

1. Örnek: Tüm Veri Bloklarını Sıfırlama ve Sector Trailer Yapılandırma İşlemi

Bu örnekte, kullanıcı kart üzerindeki **bütün veri bloklarını 0x00 ile sıfırlar** ve ardından **her sector trailer bloğunu**, cihazın belleğinde önceden tanımlı **opsiyonel KeyB** ve sabit erişim bitleri (**7F 07 88 40**) kullanarak yeniden yapılandırır.

HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	KEY TYPE	AUTH TYPE	CRC HIGH	CRC LOW
0xAA	0x01	0x11	0x03	0x04	0x02	0x1B	0xAE

Giden Mesaj Açıklamaları:

- HEADER (0xAA):** Komut başlangıcını belirtir.
- DEVICE_ADDRESS (0x01):** Hedef cihaz adresi.
- COMMAND_TYPE (0x11):** Sector Trailer yönetimi komut tipi.
- SUB_COMMAND_TYPE (0x03):** Tüm veri bloklarını sıfırlayıp trailer yapılandırma alt komutu.
- KEY_TYPE (0x04):** Opsiyonel KeyB kullanılacak. (Bkz: 7.3.2.5.2)
- AUTH_TYPE (0x02):** AUTHENT1B kimlik doğrulama yöntemi kullanılacak.
- CRC_HIGH (0x1B) & CRC_LOW (0xAE):** Opsiyonel veri bütünlüğü kontrol kodu.

Gelen Yanıt

HEADER	DEVICE ADDRESS	STATUS CODE HIGH	STATUS CODE LOW	CRC HIGH	CRC LOW
0xBB	0x01	0x00	0x00	0x5C	0x3E

Gelen Yanıt Açıklamaları:

- HEADER (0xBB):** Gelen yanıt mesajının başlangıcını belirtir.
- DEVICE_ADDRESS (0x01):** Yanıtı gönderen cihazın adresi.
- STATUS_CODE_HIGH (0x00) & STATUS_CODE_LOW (0x00):** İşlemin başarılı şekilde tamamlandığını gösterir.
- CRC_HIGH (0x5C) & CRC_LOW (0x3E):** Yanıt veri bütünlüğü için opsiyonel hata kontrol alanı.

Bu İşlemden Gerçekleşenler

- Kullanıcı, cihaza **AA 01 11 03 04 02 1B AE** komutunu gönderir.
- Cihaz, kart ile fiziksel temas kurar ve AUTHENT1B doğrulama yöntemi ile tüm bloklara erişim sağlar.
- Kart üzerindeki **tüm veri blokları 0x00 ile sıfırlanır**.
- Her sector trailer bloğu, **opsiyonel KeyB** ve **7F 07 88 40** erişim bitleri kullanılarak güncellenir.
- Tüm işlemler başarıyla tamamlanırsa cihaz **0x00 0x00** status kodu ile yanıt döner.
- Eğer işlem başarısız olursa dönen hata kodu detayları için [6.4.4 RFID Kart Okuma/Yazma Hataları](#) bölümüne bakılmalıdır.

7.8.4 Kartı Tamamen Formatlama ve Fabrika Ayarlarına Döndürme

Bu komut, RFID kart üzerindeki tüm veri bloklarını sıfırlar ve her sektörün trailer bloğunu, standart fabrika ayarlarına uygun şekilde yeniden yapılandırır.

İşlem sonucunda kart, üretim çıkışındaki (default) ayarlarına geri döner.

Kart üzerindeki tüm kullanıcı verileri ve güvenlik anahtarları temizlenir ve varsayılan güvenlik yapılandırması oluşturulur.

Kullanım Amaçları

- Kartı tamamen temizleyerek yeni bir kullanım için hazırlamak.
- Tüm eski kullanıcı verilerini, erişim kontrol yapılarını ve özel anahtarları kaldırmak.
- Kartı üretim çıkışı (fabrika ayarı) yapılandırmasına geri döndürmek.

İşlem Sonucunda Kartın Yapısı

Alan	İşlem Sonrası Durumu
Data Blokları	Tüm veri blokları 0x00 olarak sıfırlanır.
Key A	Her sektörde FF FF FF FF FF FF olarak ayarlanır.
Access Bits	Her sektörde FF 07 80 69 olarak sabitlenir.
Key B	Her sektörde FF FF FF FF FF FF olarak ayarlanır.

Tablo 46 Format İşlem Sonucunda Kartın Yapısı

Data Blocks: İçerik tamamen temizlenir (0x00).

Sector Trailer: KeyA ve KeyB alanları fabrika değerlerine (0xFF) döner.

Access Bits: Standart erişim hakları (FF 07 80 69) yazılır.

Gönderilen Mesaj Formatı

HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	KEY TYPE	AUTH TYPE	CRC HIGH (Optional)	CRC LOW (Optional)
0xAA	0x01	0x11	0x04	0xFF	0xFF	0xFF	0xFF

Alan Açıklamaları

- **HEADER (0xAA):** Komut başlangıcını belirtir.
- **DEVICE_ADDRESS (0x01):** Hedef cihazın adresidir.
- **COMMAND_TYPE (0x11):** Sector Trailer yönetim komutları grubu.
- **SUB_COMMAND_TYPE (0x04):** Kartı tamamen formatlama ve sıfırlama alt komutu.
- **KEY_TYPE:**
 - 0x01: KeyA kullanılır.
 - 0x02: KeyB kullanılır.
 - 0x03: Opsiyonel KeyA kullanılır.
 - 0x04: Opsiyonel KeyB kullanılır.(Detaylı bilgi için bkz: [7.3.2.5.2 Kullanıcı Tarafından Tanımlanabilen Anahtarlar \(Optional KeyA ve Optional KeyB\)](#))
- **AUTH_TYPE:**
 - 0x01: AUTHENT1A kullanılarak doğrulama yapılır.
 - 0x02: AUTHENT1B kullanılarak doğrulama yapılır.
- **CRC_HIGH & CRC_LOW:** Opsiyonel veri bütünlüğü alanlarıdır.



Advanced RFID Card Reader Writer

Bu İşlemden Gerçekleşen Adımlar

1. Kullanıcı, 0xAA 0x01 0x11 0x04 formatında bir komut gönderir ve KEY_TYPE ile AUTH_TYPE alanlarını belirtir.
2. Cihaz, kartı algılar ve tüm sektörler için kimlik doğrulaması yapar.
3. Başarılı kimlik doğrulama sonrası:
Tüm **veri blokları** 0x00 ile sıfırlanır.
Tüm **sector trailer blokları** fabrika ayarlarına (FF FF FF FF FF FF, FF 07 80 69, FF FF FF FF FF FF) döndürülür.
4. İşlem başarılı olursa cihaz, "0x00 0x00" durum kodu ile yanıt verir.
5. Eğer hata oluşursa, hata detayları için [6.4.4 RFID Kart Okuma/Yazma Hataları](#) bölümüne başvurulmalıdır.

Önemli Notlar

- Kart tamamen sıfırlanır. Eski veri ve özel anahtarlar kurtarılamaz.
- İşlem geri alınamaz.
- İşlem yalnızca **MODE_ADVANCED** modda çalıştırılabilir.
- Formatlama sonrası kart, standart RFID kart yazılım ve okuma işlemleri için yeniden hazırlanır.



Advanced RFID Card Reader Writer

1. Örnek: Kartı Tamamen Formatlama ve Fabrika Ayarlarına Döndürme İşlemi

Bu örnekte kullanıcı, kart üzerindeki tüm veri bloklarını sıfırlar (0x00 yazar) ve her sector trailer bloğunu fabrika ayarlarına (KeyA, Access Bits ve KeyB) döndürür.

İşlemden opsiyonel KeyB kullanılır ve kimlik doğrulama AUTHENT1B yöntemiyle yapılır.

Gönderilen Komut

HEADER	DEVICE ADDRESS	COMMAND TYPE	SUB COMMAND TYPE	KEY TYPE	AUTH TYPE	CRC HIGH	CRC LOW
0xAA	0x01	0x11	0x04	0x04	0x02	0x9E	0x3E

Giden Mesaj Açıklamaları

- HEADER (0xAA):** Komut başlangıcını belirtir.
- DEVICE_ADDRESS (0x01):** Komutun hedeflendiği cihaz adresi.
- COMMAND_TYPE (0x11):** Sector Trailer yönetimi için ana komut tipi.
- SUB_COMMAND_TYPE (0x04):** Kartı tamamen sıfırlama ve fabrika ayarlarına döndürme alt komutu.
- KEY_TYPE (0x04):** Opsiyonel KeyB kullanılır. (Bkz: 7.3.2.5.2)
- AUTH_TYPE (0x02):** AUTHENT1B kimlik doğrulama yöntemi kullanılır.
- CRC_HIGH (0x9E) & CRC_LOW (0x3E):** Veri bütünlüğü sağlamak için kullanılan opsiyonel hata kontrol kodlarıdır.

Gelen Yanıt

HEADER	DEVICE ADDRESS	STATUS CODE HIGH	STATUS CODE LOW	CRC HIGH	CRC LOW
0xBB	0x01	0x00	0x00	0x5C	0x3E

Gelen Yanıt Açıklamaları

- HEADER (0xBB):** Yanıt mesajının başlangıcını belirtir.
- DEVICE_ADDRESS (0x01):** Yanıtı gönderen cihazın adres bilgisidir.
- STATUS_CODE_HIGH (0x00) & STATUS_CODE_LOW (0x00):** İşlemin başarıyla tamamlandığını gösterir.
- CRC_HIGH (0x5C) & CRC_LOW (0x3E):** Yanıt veri bütünlüğü için eklenmiş opsiyonel hata kontrol alanıdır.

Bu İşlemden Gerçekleşenler

- Kullanıcı, cihaza **AA 01 11 04 04 02 9E 3E** komutunu gönderir.
- Cihaz, kartı algılar ve AUTHENT1B yöntemiyle kimlik doğrulama yapar.
- Kart üzerindeki tüm veri blokları 0x00 ile sıfırlanır.
- Her sector trailer bloğu:
 - KeyA:** FF FF FF FF FF FF
 - Access Bits:** FF 07 80 69
 - KeyB:** FF FF FF FF FF FF olacak şekilde fabrika ayarlarına geri döndürülür.
- İşlem başarılı ise cihaz **0x00 0x00** status kodu ile yanıt verir.
- Eğer hata oluşursa, detaylar için [6.4.4 RFID Kart Okuma/Yazma Hataları](#) bölümüne başvurulmalıdır.