# Dokumentacja techniczna

*Czytnik RFID*

# PAC-PUG
# PAC-PUB

wersja dokumentacji:PAC-PU-MAN-V2
obowiązuje od wersji firmware PAC-PU-v3.0


*PAC-PUG*

Wprowadzenie

PAC-PUx jest ladowym czytnikiem kart RFID z rodziny Mifare.

Posiada on następującą funkcjonalność:

- Obsługuje transpondery: Mifare S50, Mifare S70, Mifare UltraLight, Mifare DesFire, Mifare UltraLight C, Mifare Plus S, Mifare Plus X
- Interfejs USB (urządzenie kompozytowe)
  - w klasie CDC (emulacja portu szeregowego)
  - w klasie HID (emulacja klawiatury)
- Wbudowany buzzer,
- Wbudowane dwie diody LED dowolnego przeznaczenia oraz dioda sygnalizująca zasilanie
- Wbudowany przycisk powrotu do ustawień fabrycznych
- Odczytywanie dwustanowego wejścia
- Możliwość pełnego dostępu do wszystkich sektorów kart Mifare na poziomie odczytu i zapisu.
- Wbudowany mechanizm MAD (Mifare Application Directory)
- Dane zabezpieczone hasłem
- Aktualizacja oprogramowania poprzez interfejs USB

# 1 Dane techniczne

| Obsługiwana funkcjonalność w zależności od typu transpondera / karty: | | |
|---|---|---|
| Typ karty mifare | Odczyt numeru ID | Pełny zapis i odczyt bloków pamięci |
| S50 | TAK | TAK |
| S70 | TAK | TAK |
| UltraLight | TAK | TAK |
| DesFire | TAK | TAK |
| Mifare Plus | TAK | TAK (SL1,SL3) |

| Parametry czytnika PAC-PUx | |
|---|---|
| Napięcie zasilania | 5 V(USB) |
| Maksymalny prąd zasilania | 200 mA |
| Znamionowa częstotliwość RF pracy modułu | 13,56 MHz |
| Odległość odczytu transponderów | do 7 cm |
| Wymiary(szer.* dł. * wys.) | 92x146x29 |
| USB | Klasa CDC:<br>2400, 4800, 9600, 19200, 38400, 57600, 115200 b/s,<br>8 bitów danych, 1 bit stopu, bez bitu parzystości,<br>Zgodna z „Protokołem Netronix”<br>Klasa HID:<br>Emulacja klawiatury |
| Temperatura pracy | 0-50st.C |

# 2 Rozkazy transmisji szeregowej

Czytnik PAC-PUx widziany jest przez PC jako wirtualny port szeregowy.
W niniejszej dokumentacji opis protokołu ograniczony został do opisu rozkazów i odpowiedzi oraz ich parametrów. Nagłówek oraz suma kontrolna CRC występuje zawsze i jest zgodna z pełną dokumentacją "Protokół Netronix".

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_NazwaRozkazu | Parametry_rozkazu1…n | CRC |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_NazwaRozkazu +1 | Parametry_odpowiedzi1…m | KodOperacji | CRC |

Pracę z protokołem NETRONIX przetestować można za pomocą narzędziowego, darmowego oprogramowania „FRAMER". http://www.netronix.pl/software/oprogramowanie/framer.html

## 2.1 Zarządzanie kluczami

Zarządzanie kluczami sprowadza się do zapisu kluczy do wewnętrznej pamięci kluczy. Kluczy tych w celach bezpieczeństwa nie można odczytać. Istnieją dwa obszary pamięci, osobno dla kluczy kart Mifare Classic, osobno dla kluczy AES128bits i 3DES.

W celu utrzymania najwyższego bezpieczeństwa danych istnieje pewna poprawna filozofia pracy z kluczami.
Polega ona na zapisie kluczy przez jednostki lub osoby posiadające najwyższy stopień zaufania. Taki zapis odbywa się tylko raz lub bardzo rzadko.
Praca czytnika w konkretnej aplikacji polega nie na używaniu klucza wprost ale na wywoływaniu odpowiedniego numeru klucza w celu zalogowania się do sektora.
W ten sposób w konkretnej aplikacji klucz w zasadzie nie pojawia się na magistrali danych.

Dodatkowo użytkownik powinien zadbać aby klucz miał odpowiednie prawa dostępu do sektorów. Realizuje się to poprzez proces inicjalizacji kart, gdzie zapisuje się do kart nowe tajne klucze wraz z odpowiednimi prawami dostępu przydzielonymi tym kluczom.

Każdemu sektorowi transpondera przyporządkowany jest klucz A i klucz B.
Komendy C_LoadKeyToSKB oraz C_LoadKeyToDKB zapisują klucze Mifare Classic do pamięci czytnika bez informacji jakiego rodzaju jest to klucz (A czy B). Komenda C_DesSaveKey służy do zapisu klucza 3DES/AES (szczegóły w rozdziale obsługa Mifare Plus)
Użytkownik podczas logowania do sektora musi podać jako parametr 0xAA lub 0xBB jeżeli chce aby wywołany klucz był traktowany jako A lub jako B.

### 2.1.1 Zapis klucza MIFARE Classic do dynamicznej pamięci kluczy

Pamięć dynamiczna charakteryzuje się samoczynnym kasowaniem jej zawartości w przypadku zaniku zasilania. Jej zawartość można wielokrotnie nadpisywać.

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_LoadKeyToDKB | Key1…6 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_LoadKeyToDKB | Zapis klucza do dynamicznej pamięci kluczy | 0x14 |
| Key1…6 | 6 bajtowy klucz | dowolne |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_LoadKeyToDKB +1 | | KodOperacji | CRC |

### 2.1.2 Zapis klucza MIFARE Classic do statycznej pamięci kluczy

Pamięć statyczna charakteryzuje się nie kasowaniem jej zawartości w przypadku zaniku zasilania. Jej zawartość można wielokrotnie nadpisywać.

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_LoadKeyToSKB | Key1…6, KeyNo | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_LoadKeyToSKB | Zapis klucza do statycznej pamięci kluczy | 0x16 |
| Key1…6 | 6 bajtowy klucz | dowolne |
| KeyNo | Numer klucza.<br>W czytniku można zapisać do 32 różnych kluczy. | 0x00…0x1f |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_LoadKeyToSKB +1 | | KodOperacji | CRC |

### 2.1.3 Zapis klucza AES / 3DES do statycznej pamięci kluczy

Pamięć statyczna charakteryzuje się nie kasowaniem jej zawartości w przypadku zaniku zasilania. Jej zawartość można wielokrotnie nadpisywać.

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesSaveKey | KeyNo, Key0..Key15 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_ DesSaveKey | Zapis klucza do statycznej pamięci kluczy | 0x38 |
| KeyNo | Numer klucza.<br>W czytniku można zapisać do 32 różnych kluczy. | 0x00…0x1f |
| Key0..Key15 | 16-bajtowy klucz | |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_ DesSaveKey +1 | | KodOperacji | CRC |

## 2.2 Rozkazy wspólne do komunikacji z transponderami

### 2.2.1 Załączanie i wyłączanie pola czytnika

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_TurnOnAntennaPower | State | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_TurnOnAntennaPower | Załączanie i wyłączanie pola czytnika | 0x10 |
| State | stan załączenia | 0x00 – wyłączanie pola<br>0x01 – załączanie pola |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_TurnOnAntennaPower +1 | | KodOperacji | CRC |

### 2.2.2 Wyselekcjonowanie jednego transpondera z wielu

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_Select | RequestType | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_Select | Wyselekcjonowanie jednego transpondera z wielu | 0x12 |
| RequestType | sposób selekcjonowania transpondera | 0x00 - Standardowe selekcjonowanie transponderów z grupy tych nie będących w uśpieniu<br>0x01 - Selekcjonowanie transponderów z grupy wszystkich będących w polu czytnika. |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_Select +1 | ColNo, CardType, ID1…….IDn | KodOperacji | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | znaczenie |
|---|---|---|
| ColNo | Ilość kolizji podczas selekcjonowania jednego transpondera.<br>Liczba ta może świadczyć ile nie uśpionych transponderów jednocześnie jest w polu. | |
| CardType | Typ wyselekcjonowanego transpondera | 0x50 – S50<br>0x70 – S70<br>0x10 – Ultra Light<br>0xdf – Des Fire |
| ID1…IDn | Unikalny numer transpondera | ID1 – LSB,<br>IDn – MSB |

### 2.2.3 Uśpienie transpondera będącego w polu

Aby uśpić transponder, musi być on wcześniej wyselekcjonowany.

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_Halt | | CRC |

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_Halt | Uśpienie transpondera będącego w polu | 0x40 |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_Halt+1 | | KodOperacji | CRC |

## 2.3 Rozkazy do komunikacji z transponderami Mifare Classic

### 2.3.1 Logowanie do sektora transpondera za pomocą Dynamicznego Klucza

Aby logowanie zakończyło się powodzeniem konieczne jest po każdym załączeniu czytnika, ponowne załadowanie Dynamicznego Bufora Klucza.

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_LoginWithDKB | SectorNo, KeyType, DKNo | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_LoginWithDKB | Logowanie do sektora | 0x18 |
| SectorNo | Numer sektora transpondera do którego użytkownik chce się zalogować | **NumeracjaBlokówISektorów |

| | | |
|---|---|---|
| KeyType | Typ klucza, jaki zawarty jest w wewnętrznym Dynamicznym Buforze Klucza | 0xAA –klucz typu A<br>0xBB – klucz typu B |
| DKNo | Numer dynamicznego klucza | 0x00 |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_LoginWithDKB +1 | | KodOperacji | CRC |

### 2.3.2 Logowanie do sektora transpondera za pomocą Statycznego Bufora Kluczy

Aby logowanie zakończyło się powodzeniem konieczne jest wcześniejsze załadowanie Statycznego Bufora Kluczy.

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_LoginWithSKB | SectorNo, KeyType, SKNo | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_LoginWithSKB | Logowanie do sektora | 0x1a |
| SectorNo | Numer sektora transpondera do którego użytkownik chce się zalogować | **NumeracjaBlokówISektorów |
| KeyType | Typ klucza, jaki zawarty jest w wewnętrznym Dynamicznym Buforze Klucza | 0xAA –klucz typu A<br>0xBB – klucz typu B |
| SKNo | Numer statycznego klucza | 0x00…0x1F |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_LoginWithSKB +1 | | KodOperacji | CRC |

### 2.3.3 Odczyt zawartości bloku transpondera

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_ReadBlock | BlockNo | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_ReadBlock | Odczyt zawartości bloku transpondera | 0x1e |
| BlockNo | Numer bloku w ramach danego sektora | **NumeracjaBlokówISektorów |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_ReadBlock +1 | Data1….. Data16 | KodOperacji | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| Data1…. Data16 | Dane odczytane z bloku transpondera | |

### 2.3.4 Zapis zawartości bloku transpondera

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_WriteBlock | BlockNo, Data1….. Data116 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_WriteBlock | Zapis zawartości bloku transpondera | 0x1c |
| BlockNo | Numer bloku w ramach danego sektora | **NumeracjaBlokówISektorów |
| Data1…. Data16 | Dane jakie mają być zapisane w bloku transpondera | dowolne |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_WriteBlock +1 | | KodOperacji | CRC |

### 2.3.5 Kopiowanie zawartości bloku transpondera do innego bloku

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_CopyBlock | SourceBlockNo, TargetBlockNo | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_CopyBlock | Kopiowanie zawartości bloku transpondera do innego bloku | 0x60 |
| SourceBlockNo | źródłowy blok | **NumeracjaBlokówISektorów |
| TargetBlockNo | docelowy blok dla danych | |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_CopyBlock +1 | | KodOperacji | CRC |

### 2.3.6 Zapis wartości do bloku transpondera

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_WriteValue | BlockNo, BackupBlockNo,Value1...4, | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_WriteValue | Zapis wartości do bloku transpondera | 0x34 |

| | | |
|---|---|---|
| BlockNo | Numer bloku w ramach danego sektora, w którym Wartość będzie zapisana | **NumeracjaBlokówISektorów |
| BackupBlockNo | Deklarowany numer bloku zawierający kopię Wartości. BackupBlockNo nie ma to istotnego znaczenia dla działania systemu a użytkownik sam może/powinien zobić kopię Wartości. | **NumeracjaBlokówISektorów |
| Value1...4 | Wartość zapisywana do bloku transpondera | dowolne |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_WriteValue +1 | | KodOperacji | CRC |

### 2.3.7 Odczyt wartości z bloku transpondera

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_ReadValue | BlockNo | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_ReadValue | Odczyt wartości z bloku transpondera | 0x36 |
| BlockNo | Numer bloku w ramach danego sektora, z którego Wartość będzie odczytana | **NumeracjaBlokówISektorów |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_ReadValue+1 | Value1...4, BackupBlockNo | KodOperacji | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| Value1...4 | Wartość odczytana z bloku transpondera | |
| BackupBlockNo | Numer bloku który może zawierać kopię Wartości | **NumeracjaBlokówISektorów |

### 2.3.8 Zwiększenie wartości zawartej w bloku transpondera

Aby wykonanie rozkazu przyniosło poprawne rezultaty w deklarowanym bloku dane muszą mieć format „Wartości”.

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_IncrementValue | BlockNo, Value1...4 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_IncrementValue | Zwiększenie wartości zawartej w bloku transpondera | 0x30 |
| BlockNo | Numer bloku w ramach danego sektora, w którym Wartość będzie modyfikowana | **NumeracjaBlokówISektorów |
| Value1...4 | wartość dodawana do istniejącej rzeczywistej wartości bloku transpondera | |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_IncrementValue +1 | | KodOperacji | CRC |

### 2.3.9 Zmniejszanie wartości zawartej w bloku transpondera

Aby wykonanie rozkazu przyniosło poprawne rezultaty w deklarowanym bloku dane muszą mieć format „Wartości”.

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DecrementValue | BlockNo, Value1...4 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DecrementValue | Zmniejszanie wartości zawartej w bloku transpondera | 0x32 |
| BlockNo | Numer bloku w ramach danego sektora, w którym Wartość będzie modyfikowana | **NumeracjaBlokówISektorów |
| Value1...4 | wartość odejmowana od istniejącej rzeczywistej wartości bloku transpondera | dowolna |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DecrementValue+1 | | KodOperacji | CRC |

## 2.4 Rozkazy do komunikacji z transponderami Mifare Ultralight, Mifare Ultralight C

### 2.4.1 Zapis zawartości strony w Mifare UL

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_WritePage4B | PageAdr, Data1...4 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_WritePage4B | Zapis zawartości strony w Mifare UL | 0x26 |
| PageAdr | Numer strony w transponderze | 0x00…0x0f |
| Data1...4 | Dane jakie mają być zapisane | dowolne |
| | | |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_WritePage4B +1 | | KodOperacji | CRC |

### 2.4.2 Odczyt zawartości stron w Mifare UL

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_ReadPage16B | PageAdr | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_ReadPage16B | Odczyt zawartości stron w Mifare UL | 0x28 |
| PageAdr | Adres strony począwszy od której powinien rozpocząć się odczyt 4 kolejnych stron. Jeżeli PageAdr>0x???? to nastąpi odczyt stron znajdujących się na początku pamięci. | 0x00…0x0f |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_ReadPage16B +1 | Data1…16 | KodOperacji | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| Data1…16 | Odczytane dane z 4 kolejnych stron. | dowolne |

### 2.4.3 Uwierzytelnienie dla transpondera Ultralight C

Uwaga! Uwierzytelnienie jest możliwe tylko po uprzednim zapisaniu kluczy w pamięci transpondera.

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_ULC_Auth | KeyIdx | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_ULC_Auth | | 0x3C |
| KeyIdx | Indeks klucza zapisanego w czytniku | 0x00…0x1f |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_ ULC_Auth +1 | | KodOperacji | CRC |

## 2.5 Rozkazy do komunikacji z transponderami Mifare Plus

### 2.5.1 Rozkazy poziomu SL0

#### 2.5.1.1 Write Perso –inicjalizacja karty

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_MfPlusCMD | 0xA8, AdrH, AdrL, Data{0..15} | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_MfPlusCMD | Komenda obsługi MFPlus | 0x3A |
| 0xA8 | Podkomenda ‘Write Perso’ | |
| AdrH, AdrL | Dwubajtwtowy numer bloku lub klucza do zapisania | Zgodnie z dokumentacją Transpondera MFPLUS |
| Data{0..15} | Klucz lub dane do zapisania | dowolne |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_ MfPlusCMD +1 | | KodOperacji | CRC |

#### 2.5.1.2 Commit Perso – przejście do następnego poziomu SL

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_MfPlusCMD | 0xAA | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_MfPlusCMD | Komenda obsługi MFPlus | 0x3A |
| 0xAA | Podkomenda ‘Commit Perso’ | |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_ MfPlusCMD +1 | | KodOperacji | CRC |

### 2.5.2 Rozkazy poziomu SL1

W tym poziomie transponder Mifare Plus jest kompatybilny z transponderem Mifare Classic. Dostępne są wszystkie komendy związane z obsługą Mifare Classic, dodatkowo wprowadzona została funkcjonalność uwierzytelniania AES

#### 2.5.2.1 Uwierzytelnienie SL1

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_MfPlusCMD | 0x10, KeyIdx | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_MfPlusCMD | Komenda obsługi MFPlus | 0x3A |
| 0x10 | Podkomenda ‘Authentication SL1’ | |
| KeyIdx | Indeks klucza AES zapisanego w czytniku | 0x00-0x1F |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_ MfPlusCMD +1 | | KodOperacji | CRC |

#### 2.5.2.2 Przejście na wyższy poziom SL/ sprawdzenie oryginalności transpondera

Przejście na wyższy poziom SL lub sprawdzenie oryginalności następuje po poprawnej autoryzacji AES z odpowiednim identyfikatorem klucza

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_MfPlusCMD | 0x70, AdrH, AdrL, KeyIdx | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_MfPlusCMD | Komenda obsługi MFPlus | 0x3A |
| 0x70 | Podkomenda ‘First Auth’ | |
| AdrH, AdrL | Dwubajtwtowy numer bloku lub klucza do zapisania | 0x9002 – przejście do poziomu SL2<br>0x9003 – przejście do poziomu SL3<br>0x8000 – sprawdzenie oryginalności transpondera |
| KeyIdx | Indeks klucza AES zapisanego w czytniku | 0x00-0x1F |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_ MfPlusCMD +1 | | KodOperacji | CRC |

### 2.5.3 Rozkazy poziomu SL3

#### 2.5.3.1 Wprowadzenie transpondera w tryb ISO14443-4

Każda komenda związana z SL3 musi być poprzedzona jednorazowym wprowadzeniem transpondera w tryb zgodności z ISO14443-4

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_Init_ISO14443-4 | CID | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_Init_ISO14443-4 | | 0x62 |
| CID | Identyfikator CID | 0x00 |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_Init_ISO14443-4+1 | | KodOperacji | CRC |

### 2.5.3.2 Logowanie do sektora

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_MfPlusCMD | 0x1A, Sector, KeyType, KeyIdx | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_MfPlusCMD | Komenda obsługi MFPlus | 0x3A |
| 0x1A | Podkomenda ‘sector login’ | |
| Sector | Numer sektora | 0x00-0x1f – karta Plus 2K<br>0x00-0x27 – karta Plus 4k |
| KeyType | Typ klucza | 0xAA – klucz A<br>0xBB – klucz B |
| KeyIdx | Indeks klucza AES zapisanego w czytniku | 0x00-0x1F |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_ MfPlusCMD +1 | | KodOperacji | CRC |

### 2.5.3.3 Odczyt zawartości bloku transpondera

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_ MfPlusCMD | read_cmd, block | CRC |

Gdzie:

<table>
<tr><th>Nazwa parametru</th><th colspan="4">Opis parametru</th><th>Zakres wartości</th></tr>
<tr><td>C_MfPlusCMD</td><td colspan="4">Komenda obsługi MFPlus</td><td>0x3A</td></tr>
<tr><td>read_cmd</td><td colspan="4">Typ procedury odczytu:<br><table><tr><th>cmd.</th><th>MAC on command</th><th>MAC on resonse</th><th>Plain /encrypted</th></tr><tr><td>0x30</td><td>Yes</td><td>No</td><td>Encrypted*</td></tr><tr><td>0x31</td><td>Yes</td><td>Yes</td><td>Encrypted*</td></tr><tr><td>0x32</td><td>Yes</td><td>No</td><td>Plan</td></tr><tr><td>0x33</td><td>Yes</td><td>Yes</td><td>Plan</td></tr></table></td><td>0x30-0x33</td></tr>
<tr><td>block</td><td colspan="4">Numer bloku do odczytu</td><td>0-3 dla sektorów<32<br>0-15 dla sektorów>32</td></tr>
</table>

*tylko transpondery Plus X

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_ MfPlusCMD +1 | Data1….. Data16 | KodOperacji | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| Data1…. Data16 | Dane odczytane z bloku transpondera | |

### 2.5.3.4 Zapis zawartości bloku transpondera

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_ MfPlusCMD | write_cmd, block, data0..data15 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_MfPlusCMD | Komenda obsługi MFPlus | 0x3A |
| write _cmd | Typ procedury zapisu: | 0xA0-0xA3 |
| block | Numer bloku do odczytu | 0-3 dla sektorów<32<br>0-15 dla sektorów>32 |
| data0..data15 | Dane do zapisu bloku transpondera | |

| cmd. | MAC on command | MAC on resonse | Plain /encrypted |
|---|---|---|---|
| 0xA0 | Yes | No | Encrypted* |
| 0xA1 | Yes | Yes | Encrypted* |
| 0xA2 | Yes | No | Plain |
| 0xA3 | Yes | Yes | Plain |

*tylko transpondery Plus X

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_ MfPlusCMD +1 | | KodOperacji | CRC |

### 2.5.4 Czasy trwania operacji dla Mifare Plus

Poniższe zestawienie określa czas trwania poszczególnych operacji, liczony od momentu zakończenia wysyłania ramki komendy (RS) do momentu rozpoczęcia wysyłania ramki odpowiedzi(RS)

| Operacja | Rezultat poprawny [ms] | Rezultat niepoprawny [ms] |
|---|---|---|
| SELECT | 14 | 12 |
| LOGIN SL3 | 25 | 100 |
| READ BLOCK | 10 | 100 |
| WRITE BLOCK | 13 | 100 |

## 2.6 Obsługa transponderów DESFire, DESFire EV1

### 2.6.1 Autoryzacja, logowanie do aktualnie wyselekcjonowanej aplikacji

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesAuth (0x42) | KeyNo{0..0x10}, KeyIdx, AuthType | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesAuth | Komenda autoryzacji | 0x42 |
| KeyNo | Numer klucza w odniesieniu do transpondera | 0x00..0x10 |
| KeyIdx | Indeks klucza AES zapisanego w czytniku | 0x00-0x1F |
| AuthType | Typ autoryzacji :<br>0x0A – DES<br>0xAA - AES | 0x0A, 0xAA |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesAuth +1 | | KodOperacji | CRC |

### 2.6.2 Zmiana ustawień klucza Master aktualnie wybranej aplikacji

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesChangeKeySett (0x44) | KeySettings | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesChangeKeySett | Komenda zmiany ustawień klucza | 0x44 |
| KeySettings | Bajt konfiguracyjny | 0x00..0x0f |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesChangeKeySett+1 | | KodOperacji | CRC |

Struktura bajtu konfiguracyjnego *KeySettings*:

| Bit | Znaczenie |
|---|---|
| 0 | 0 – klucz PICC Master key jest niemodyfikowalny |
| | 1* – klucz PICC Master key jest modyfikowalny |
| 1 | 0 – wywołanie funkcji C_DesGetAppIDs wymaga autoryzacji z użyciem PICC Master key |
| | 1* – wywołanie funkcji C_DesGetAppIDs nie wymaga autoryzacji |
| 2 | 0 – utworzenie/usunięcie aplikacji wymaga autoryzacji z użyciem PICC Master key |
| | 1* -utworzenie nowej aplikacji nie wymaga autoryzacji,<br>usunięcie aplikacji wymaga autoryzacji kluczem danej aplikacji lub PICC Master key |
| 3 | 0 – zmiana konfiguracji klucza PICC Master Key jest niemożliwa |
| | 1* - zmiana konfiguracji klucza PICC Master Key dozwolona w przypadku autoryzacji z użyciem tego klucza |
| 4 | RFU – 0 |
| 5 | RFU – 0 |
| 6 | RFU – 0 |
| 7 | RFU – 0 |

* - ustawienie domyślne

### 2.6.3 Zmiana klucza

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesChangeKey (0x46) | KeyNo, NewEESavedKey,[PrevEESavedKey] | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesChangeKey | Komenda zmiany klucza | 0x46 |
| KeyNo | Numer klucza do zmiany | 0x00..0x0D |
| NewEESavedKey | Indeks nowego klucza zapisanego w pamięci czytnika | 0x00..0x13 |
| PrevEESavedKey | • Jeśli zmieniany klucz nie jest tym, którym nastąpiła aktualna autoryzacja, podajemy indeks aktualnego klucza, który będzie zmieniany<br>• Jeśli zmieniany klucz jest tym samym, którym nastąpiła aktualna autoryzacja, parametr ten pozostawiamy pusty | 0x00..0x13 |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesChangeKey+1 | | KodOperacji | CRC |

### 2.6.4 Tworzenie aplikacji

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesCreateApp (0x48) | AId1..3,KeySettings1, KeySettings2 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesCreateApp | Komenda tworzenia aplikacji | 0x48 |
| AId1..3 | 3-bajtowy identyfikator aplikacji | 0x00..0xFF |
| KeySettings1 | Bajt konfiguracyjny (patrz poniżej) | 0x00..0x0F |
| KeySettings2 | *Bit3..bit0:*<br>Liczba kluczy przypisanych do danej aplikacji<br>*Bit7..Bit6*:<br>00 – autoryzacja DES dla całej aplikacji<br>10- autoryzacja AES dla całej aplikacji | 0x00..0x0D |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesCreateApp +1 | | KodOperacji | CRC |

Struktura bajtu konfiguracyjnego *KeySettings*:

| Bit | Znaczenie |
|---|---|
| 0 | 0 – klucz Application Master key jest niemodyfikowalny |
| | 1* – klucz Application Master key jest modyfikowalny, wymaga autoryzacji z użyciem dotychczasowego klucza AppMasterKey |
| 1 | 0 – wywołanie funkcji C_DesGetAppIDs wymaga autoryzacji z użyciem PICC Master key |
| | 1* – wywołanie funkcji C_DesGetAppIDs nie wymaga autoryzacji |
| 2 | 0 – utworzenie/usunięcie pliku wymaga autoryzacji z użyciem AppMasterKey |
| | 1* -utworzenie/usunięcie pliku nie wymaga autoryzacji z użyciem AppMasterKey |
| 3 | 0 – zmiana konfiguracji klucza Application Master Key jest niemożliwa |
| | 1* - zmiana konfiguracji klucza Application Master Key dozwolona w przypadku autoryzacji z użyciem tego klucza |
| 4 | Bit7-Bit4: określają prawa do zmian parametrów klucza |
| 5 | 0x0*:Klucz Master aplikacji jest niezbędny do zmiany ustawień kluczy |
| 6 | 0x1-0xD : autoryzacja przy pomocy klucza z tym indeksem jest konieczna do zmiany ustawień kluczy |
| 7 | 0xE :zmiana ustawień klucza wymaga autoryzacji z użyciem tego samego klucza |

* - ustawienie domyślne

### 2.6.5 Usuwanie aplikacji

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesDeleteApp (0x4a) | AId1..3 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesDeleteApp | Komenda usuwania aplikacji | 0x4a |
| AId1..3 | 3-bajtowy identyfikator aplikacji | 0x00..0xFF |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesCreateApp +1 | | KodOperacji | CRC |

### 2.6.6 Pobieranie listy aplikacji

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesGetAppIDs (0x4c) | | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesGetAppIDs | Komenda pobierania listy aplikacji | 0x4c |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesGetAppIDs +1 | N*{Aid3,Aid2,Aid1} | KodOperacji | CRC |

Zwracana jest lista numerów Aid, aktualnie istniejących aplikacji

### 2.6.7 Wybór aplikacji

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesSelectApp (0x4e) | Aid1..3 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesSelectApp | Komenda wyboru aplikacji | 0x4e |
| Aid1..3 | 3 bajtowy identyfikator aplikacji | 0x00-0xff |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesSelectApp+1 | | KodOperacji | CRC |

### 2.6.8 Formatowanie transpondera

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesFormatPICC (0x60) | | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesFormatPICC | Komenda formatowania transpondera | 0x60 |

Wykonanie tej komendy wymaga autoryzacji z użyciem klucza PICC Master key

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesFormatPICC +1 | | KodOperacji | CRC |

### 2.6.9 Inicjalizacja protokołu transmisji z transponderami DESFire

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesInitProtocol (0x62) | CID | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesInitProtocol | Komenda formatowania transpondera | 0x62 |
| CID | Logiczny numer wyselekcjonowanego transpondera | 0x00-0x0E |

Komenda ta musi wystąpić bezpośrednio po wyselekcjonowaniu transpondera komendą C_Select. Obecna wersja czytnika pozwala na pracę z jednym transponderem Desfire jednocześnie. Numer logiczny CID nie ma aktualnie znaczenia, zaleca się podawanie numeru 0

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesInitProtocol +1 | | KodOperacji | CRC |

### 2.6.10 Pobieranie listy plików aktualnie wybranej aplikacji

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesGetFileIDs (0x64) | | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesGetFileIDs | Komenda pobierania listy plików | 0x64 |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesGetAppIDs +1 | N*FileNo | KodOperacji | CRC |

Zwracana jest lista numerów plików aktualnie istniejących w wybranej aplikacji

### 2.6.11 Pobieranie właściwości pliku

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesGetFileSett (0x66) | FileNo | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesGetFileSett | Komenda pobierania właściwości pliku | 0x66 |
| FileNo | Identyfikator pliku | 0x00-0x0f |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesGetAppIDs +1 | File params... | KodOperacji | CRC |

W zależności od typu pliku zwracana jest informacja w następującym formacie:

- Dla plików *Standard Data Files* i *Backup Data Files*

| 1 byte | 1 byte | 2 bytes | 3 bytes |
|---|---|---|---|
| File type | Comm. Sett. | Access right | File size |
| | | LSB MSB | LSB MSB |

- Dla plików *Value Files* (ten typ aktualnie nie jest zaimplementowany)

| 1 byte | 1 byte | 2 bytes | 4 bytes | 4 bytes | 4 bytes | 1 byte |
|---|---|---|---|---|---|---|
| File type | Comm. Sett. | Access right | Lower limit | Upper limit | Limited credit value | Limited credit enable |
| | | LSB MSB | LSB MSB | LSB MSB | LSB MSB | |

- Dla plików *Linear/Cyclic record files*

| 1 byte | 1 byte | 2 bytes | 3 bytes | 3 bytes | 3 bytes |
|---|---|---|---|---|---|
| File type | Comm. Sett. | Access right | Record size | Maximum number of records | Current number of records |
| | | LSB MSB | LSB MSB | LSB MSB | LSB MSB |

### 2.6.12 Tworzenie plików typu *Standard Data Files*

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesCreateSTDataFile (0x68) | FileNo,ComSett,AccRight1..2,FileSize1..3 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesCreateSTDataFile | Komenda tworzenia pliku STD | 0x68 |
| FileNo | Identyfikator pliku | 0..0x0F |
| ComSett | Typ transmisji:<br>0x01 – nieszyfrowana<br>0x03 – szyfrowana DES | 0x00,0x03 |
| AccRight1..2 | Prawa dostępu do pliku, patrz tabela poniżej | 0x00..0xff |
| FileSize1..3 | 3 bajtowa wielkość pliku w bajtach,<br>w kolejności LSB..MSB | 0x00-0xff |

Bajty określające prawa dostępu:

| 15 12 | 11 8 | 7 4 | 3 0 |
|---|---|---|---|
| **Read Access** | **Write Access** | **Read & Write Access** | **Change Right Access** |
| MBS ***1st byte*** | | ***2nd byte*** | LSB |

Dwa bajty praw dostępu podzielone są na 4 pola 4 bitowe. Każde pole może zawierać wartości z przedziału 0x0 – 0xF

- Wartości z przedziału 0x0 – 0xD określają numer klucza, który będzie miał prawa do wykonania danej operacji,
- Wartość 0xE oznacza, że dana operacja nie wymaga autoryzacji
- Wartość 0xF oznacza, że nie ma dostępu do danej operacji, bez względu na użyty klucz

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesCreateSTDataFile +1 | | KodOperacji | CRC |

### 2.6.13 Tworzenie plików typu *Backup Data Files*

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesCreateBACKDataFile (0x6a) | FileNo,ComSett,AccRight1..2,FileSize1..3 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesCreateBACKDataFile | Komenda tworzenia pliku BACKUP | 0x6a |
| FileNo | Identyfikator pliku | 0..0x07 |
| ComSett | Typ transmisji:<br>0x01 – nieszyfrowana<br>0x03 – szyfrowana DES | 0x00,0x03 |
| AccRight1..2 | Prawa dostępu do pliku | 0x00..0xff |
| FileSize1..3 | 3 bajtowa wielkość pliku w bajtach<br>w kolejności LSB..MSB | 0x00-0xff |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesCreateBACKDataFile +1 | | KodOperacji | CRC |

Prawa dostępu określa się identycznie jak w przypadku plików *Standard Data Files*

Zapis pliku typu *Backup Data file* musi zakończyć się wydaniem komendy C_DesCommit.

### 2.6.14 Tworzenie plików typu Linear/Cyclic Record Files

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesCreateRecordFile (0x6c) | FileNo, ComSett, AccRight1..2, RecSize1..3,<br>RecNumb1..3, Cy/Li{0x0C,0x01} | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesCreateRecordFile | Komenda tworzenia pliku typu *Record File* | 0x6c |
| FileNo | Identyfikator pliku | 0..0x0F |
| ComSett | Typ transmisji:<br>0x01 – nieszyfrowana<br>0x03 – szyfrowana DES | 0x00,0x03 |
| AccRight1..2 | Prawa dostępu do pliku | 0x00..0xff |
| RecSize1..3 | 3 bajtowy rozmiar rekordu w bajtach,<br>w kolejności LSB..MSB | 0x00-0xff |
| RecNumb1..3 | 3 bajtowy parametr określający ilość rekordów,<br>kolejność LSB..MSB | |
| Cy/Li | 0x0c- typ cykliczny<br>0x0l – typ liniowy | 0x0C,0x01 |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesCreateRecordFile+1 | | KodOperacji | CRC |

Prawa dostępu określa się identycznie jak w przypadku plików *Standard Data Files*

### 2.6.15 Usuwanie pliku

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesDeleteFile (0x6e) | FileNo | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesDeleteFile | Komenda usuwania pliku | 0x6e |
| FileNo | Identyfikator pliku | 0x00..0x0F |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesDeleteFile+1 | | KodOperacji | CRC |

### 2.6.16 Zmiana ustawień pliku

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesChangeFileSett (0x80) | FileNo, ComSett, AccRight1..2 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesChangeFileSett | Komenda zmiany właściwości pliku | 0x80 |
| FileNo | Identyfikator pliku | 0..0x0F |
| ComSett | Typ transmisji:<br>0x01 – nieszyfrowana<br>0x03 – szyfrowana DES | 0x00,0x03 |
| AccRight1..2 | Prawa dostępu do pliku | 0x00..0xff |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesChangeFileSett+1 | | KodOperacji | CRC |

Prawa dostępu określa się identycznie jak w przypadku tworzenia plików *Standard Data Files*

### 2.6.17 Odczyt danych z pliku typu *Std/Back Data File*

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesReadData (0x82) | FileNo, Offset1..3, Length1..3 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|

| | | |
|---|---|---|
| C_DesReadData | Komenda odczytu z pliku | 0x82 |
| FileNo | Identyfikator pliku | 0..0x0F |
| Offset1..3 | 3 bajtowy parametr określający miejsce od którego zaczynamy czytać plik, kolejność LSB..MSB | 0x00-0xFF |
| Length1..3 | 3 bajtowy parametr określający ilość bajtów, które chcemy odczytać, kolejność LSB..MSB (jednorazowo odczytać można do 58 bajtów) | 0x00-0x3A |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesReadData +1 | n Bytes | KodOperacji | CRC |

### 2.6.18 Zapis danych do pliku typu *Std/Back Data File*

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesWriteData (0x84) | FileNo, Offset1..3,Data1..58 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesWriteData | Komenda zapisu do pliku | 0x84 |
| FileNo | Identyfikator pliku | 0..0x0F |
| Offset1..3 | 3 bajtowy parametr określający miejsce od którego zaczynamy zapisywać, kolejność LSB..MSB | 0x00-0xFF |
| Data1..58 | Dane, które zamierzamy zapisać do pliku, (jednorazowo zapisać można do 58bajtów) | 0x00-0xFF |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesWriteData+1 | | KodOperacji | CRC |

### 2.6.19 Zapis rekordu do pliku typu *Record Data File*

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesWriteRecord (0x86) | FileNo, Offset1..3,Data1..58 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesWriteRecord | Komenda zapisu rekordu | 0x86 |
| FileNo | Identyfikator pliku | 0..0x0F |
| Offset1..3 | 3 bajtowy parametr określający miejsce od którego zaczynamy zapisywać, kolejność LSB..MSB (wartość ta musi być mniejsza niż wielkość pojedynczego rekordu) | 0x00-0xFF |
| Data1..58 | Dane, które zamierzamy zapisać do pliku, (jednorazowo zapisać można do 58bajtów, suma tej wartości oraz offsetu musi być mniejsza niż wielkość pojedynczego rekordu) | 0x00-0xFF |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesWriteRecord+1 | | KodOperacji | CRC |

Uwaga: Zapis rekordu do pliku typu *Record File* musi zakończyć się wydaniem komendy C_DesCommit.

### 2.6.20 Odczyt rekordu z pliku typu *Record Data File*

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesReadRecord (0x88) | FileNo, WhichRecord1..3,NoOfRecords1..3 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesReadRecord | Komenda odczytu rekordu | 0x88 |
| FileNo | Identyfikator pliku | 0..0x0F |
| WhichRecord1..3 | 3 bajtowy parametr określający rekord od którego zaczynamy czytać, kolejność LSB..MSB | 0x00-0xFF |
| NoOfRecords1..3 | 3 bajtowy parametr określający ilość rekordów do przeczytania, kolejność LSB..MSB | 0x00-0xFF |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesReadRecord +1 | Record data... | KodOperacji | CRC |

Ilość odczytanych danych nie może być większa niż 58 bajtów, stąd należy zachować zasadę:
{NoOfRecords1..3}*rozmiar_rekordu < 58bytes

### 2.6.21 Czyszczenie plików typu *Record Data File*

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesClearRecordFile (0x8a) | FileNo | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesClearRecordFile | Komenda czyszczenia pliku rekordowego | 0x8a |
| FileNo | Identyfikator pliku | 0..0x0F |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesClearRecordFile+1 | | KodOperacji | CRC |

Uwaga: Operacja ta musi zakończyć się wydaniem komendy C_DesCommit.

### 2.6.22 Komenda potwierdzająca - *DesCommit*

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesCommit (0x8c) | | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesCommit | Komenda potwierdzenia | 0x8c |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesCommit+1 | | KodOperacji | CRC |

### 2.6.23 Deselekcja transpondera

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_DesDeselect (0x8e) | | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_DesDeselect | Komenda de-selekcjonująca transponder | 0x8e |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_DesDeselect+1 | | KodOperacji | CRC |

## 2.7 Transmisja danych I-Block protokołu ISO14443-4

Komenda ta umożliwia wysłanie danych do transpondera w trybie ISO14443-4, jednocześnie zwraca informacje z transpondera. Przed wykonaniem tej komendy konieczne jest przejście w tryb ISO14443-4 za pomocą komendy C_Init_ISO14443-4.

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_TranscIBlock | data | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_TranscIBlock | | 0xC8 |
| data | Dane pakietu I-Block | dowolne |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_TranscIBlock+1 | data | KodOperacji | CRC |

## 2.8 Mifare Application Directory - MAD

### 2.8.1 Formatowanie karty MAD

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_FormatMad | Type, Infobyte | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|

| | | |
|---|---|---|
| C_FormatMad<br>0xa8 | Formatowanie do MAD | 0xa8 |
| Type | 1 - MAD1 (15sektorów)<br>2 – MAD2 (30sektorów) | 0x01,0x02 |
| Infobyte | Wskaznik na sektor emitenta (domyślnie 0x00) | 0x00-0x1F |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_FormatMad+1 | | KodOperacji | CRC |

Uwagi:
Przed wykonaniem komendy C_FormatMad należy:

- wyłączyć tryb AutoReader (komendą C_SetAutoReaderConfig)
- załadować klucze (domyślnie 0xff,0xff,0xff,0xff,0xff,0xff)
- włączyć zasilanie anteny (komendą C_TurnOnAntennaPower)
- wyselekcjonować kartę (komendą C_Select)
- zalogować się do sektora nr 0 używając klucza typu AA

### 2.8.2 Dodanie aplikacji do katalogu MAD

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_AddApplication | LSB, MSB, Sector | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_AddApplication<br>0xaa | Dodanie aplikacji | 0xaa |
| LSB | mniej znaczący bajt numeru aplikacji | 0x00 - 0xFF |
| MSB | bardziej znaczący bajt numeru aplikacji | 0x00 - 0xFF |
| Sector | Numer sektora, gdzie aplikacja ma się znajdować | 0x01-0x0F :MAD1<br>0x01-0x1F :MAD2 |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_AddApplication+1 | | KodOperacji | CRC |

Uwagi:
Numer aplikacji musi być różny od 0x0000
Przed wykonaniem komendy C_AddApplication należy:

- wyłączyć tryb AutoReader (komendą C_SetAutoReaderConfig)
- załadować klucze (domyślnie 0xff,0xff,0xff,0xff,0xff,0xff)
- włączyć zasilanie anteny (komendą C_TurnOnAntennaPower)
- wyselekcjonować kartę (komendą C_Select)
- zalogować się do sektora nr 0 używając klucza typu AA

### 2.8.3 Wyszukanie sektora dla danej aplikacji

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_GetSectorMad | LSB, MSB | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_GetSectorMad | Wyszukanie sektora | 0xac |

| | | |
|---|---|---|
| 0xac | | |
| LSB | mniej znaczący bajt numeru aplikacji | 0x00 - 0xFF |
| MSB | bardziej znaczący bajt numeru aplikacji | 0x00 - 0xFF |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_GetSectorMad+1 | Sector | KodOperacji | CRC |

Uwagi:
Przed wykonaniem komendy C_GetSectorMad należy:

- wyłączyć tryb AutoReader (komendą C_SetAutoReaderConfig)
- załadować klucze (domyślnie 0xff,0xff,0xff,0xff,0xff,0xff)
- włączyć zasilanie anteny (komendą C_TurnOnAntennaPower)
- wyselekcjonować kartę (komendą C_Select)
- zalogować się do sektora nr 0 używając klucza typu AA

Jeśli bajt odpowiedzi będzie wynosił 0x00 oznacza to, że dana aplikacja nie znajduje się w katalogu MAD

### 2.8.4 Wyszukanie kolejnego sektora aplikacji

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_GetSectorMadNext | LSB, MSB | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_GetSectorMad 0xae | Wyszukanie kolejnego sektora | 0xae |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_GetSectorMadNext+1 | Sector | KodOperacji | CRC |

Uwagi:
Przed wykonaniem komendy C_GetSectorMadNext należy wykonać operację wyszukania sektora komendą C_GetSectorMad, której wynik wyszukiwania był różny od 0

Jeśli bajt odpowiedzi będzie wynosił 0x00 oznacza to, że nie znaleziono więcej sektorów dla danej aplikacji

## 2.9 Wejścia i wyjścia elektryczne

### 2.9.1 Zapis stanu wyjścia

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_WriteOutputs | IONo, State | CRC |

Gdzie:

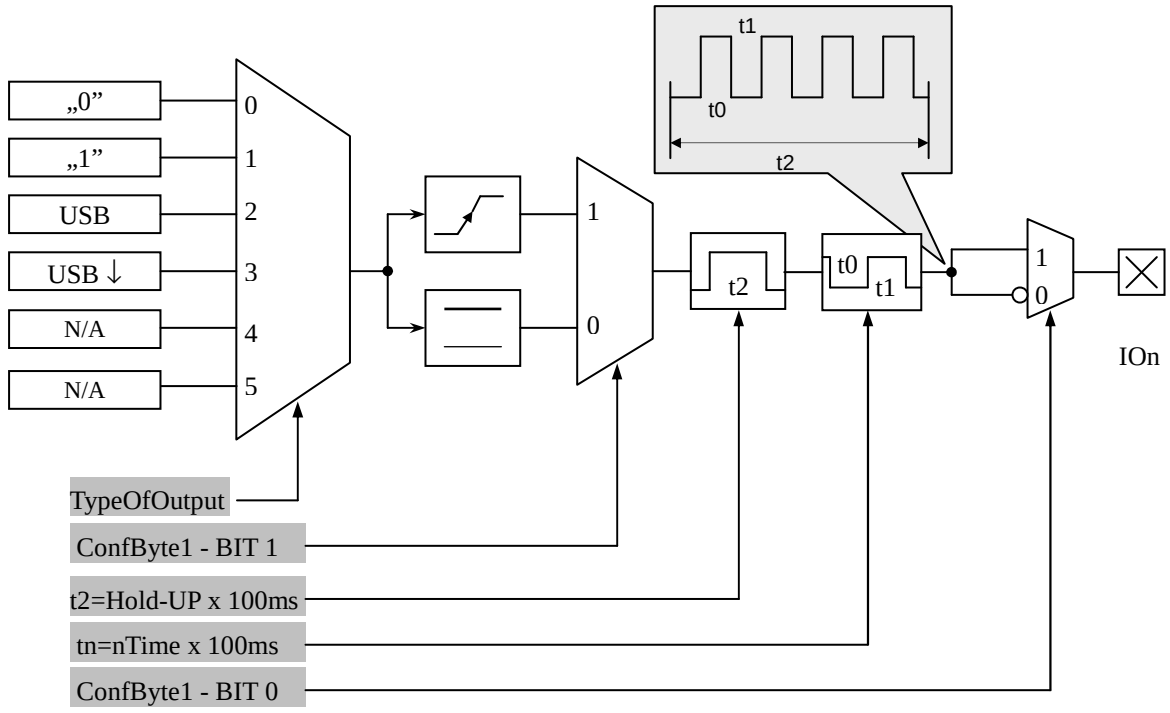
| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|

| | | |
|---|---|---|
| C_WriteOutputs | Zapis stanu wyjścia | 0x70 |
| IONo | Numer portu IO. Powinien on być skonfigurowany jako wyjście. | 0x00…0x07 |
| State | Żądany stan wyjścia | 0x00 lub 0x01 |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_WriteOutputs +1 | | KodOperacji | CRC |

### 2.9.2 Zapis konfiguracji dowolnego portu




Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_SetIOConfig | IONo, IOConfigData1…n | CRC |

Jeżeli Konfigurujemy port jako wyjście to parametry IOConfigData1…n mają postać:

Dir, ConfByte1, TypeOfOutput, Podtrzymanie, 0Time, 1Time

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_SetIOConfig | Zapis konfiguracji dowolnego portu | 0x50 |
| IONo | Numer portu IO, który ma być skonfigurowany | 0x02…0x06 |
| Dir | kierunek portu | 0x00 – wyjście |
| ConfByte1 | jeden bajt w którym najmłodszy bit określa | ConfByte1.BIT0 |

| | | |
|---|---|---|
| | typ wyjścia jako Normalnie otwarte lub Normalnie Zamknięte.<br>Następny bit określa sposób reakcji danego wyjścia jako reagujące na zmianę pobudzenia (reagujące na zbocze) lub reagujące na stan pobudzenia (reagujące na stan). | 0-Normalnie Zamknięte<br>1-Normalnie Otwarte<br><br>ConfByte1.BIT1<br>0-reaguje na poziom<br>1-reaguje na zbocze |
| TypeOfOutput | śródło sygnału sterującego | 0x00 – wyłączone na stałe<br>0x01 – załączone na stałe<br>0x02 – sterowane poprzez interface szeregowy USB<br>0x03 - sterowane poprzez interface szeregowy USB automatycznie powracające do zera |
| Podtrzymanie | Czas podtrzymania stanu załączenia po ustaniu pobudzenia. Czas ten wyrażony jest jako:<br><br>Podtrzymanie x 100ms<br><br>Podczas trwania czasu „Podtrzymanie” można skonfigurować wyjście potrafiące generować falę prostokątną. Czas jedynki i czas zera ustawiany jest następnymi parametrami: | |
| 0Time | czas logicznego zera | |
| 1Time | czas logicznej jedynki | |

Jeżeli Konfigurujemy port jako wejście to parametry IOConfigData1…n mają postać:
Dir, Triger, TypeOfInput, Opoznienie,

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_SetIOConfig | Zapis konfiguracji dowolnego portu | 0x50 |
| IONo | Numer portu IO, który ma być skonfigurowany | 0x00…0x01,0x07 |
| Dir | kierunek portu. | 0x01 – wejście |
| TypeOfInput | Typ wejścia | 0x03 |
| Opoznienie | opóźnienie | 0x00 |

Czytnik PAC_MUx nie ma możliwości przełączania kierunku portów.
W celu poprawnej konfiguracji należy dla danego portu podać poprawny kierunek.

| **SPIS ISTNIEJĄCYCH PORTÓW, KTÓRYMI MOŻNA STEROWAĆ W PAC-PU** | | |
|---|---|---|
| Numer portu | kierunek | Opis |
| 0 | wyjście | LED CZERWONY1 |

| | | |
|---|---|---|
| 1 | wyjście | LED CZERWONY2 |
| 3 | wyjście | BUZZER |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_SetIOConfig +1 | | KodOperacji | CRC |

### 2.9.3 Odczyt konfiguracji dowolnego portu

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_GetIOConfig | IONo | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_GetIOConfig | Odczyt konfiguracji dowolnego portu | 0x52 |
| IONo | Numer portu IO, który którego konfiguracja ma być odczytana | 0x00…0x07 |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_GetIOConfig +1 | IOConfigData1…n | KodOperacji | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| IOConfigData1…n | ma postać taką samą jak przy zapisie konfiguracji | |

## 2.10 Hasło dostępu

### 2.10.1 Logowanie do czytnika

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_LoginUser | Data1…n, 0x0 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_LoginUser | Logowanie do czytnika | 0xb2 |
| Data1…n | jest dowolnym łańcuchem bajtów | Dowolne z zakresu 0x01…0xff. Długość łańcucha może wynosić od 0 do 8 bajtów |
| 0x00 | Zero kończące string | 0x00 |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_LoginUser +1 | | KodOperacji | CRC |

### 2.10.2 Zmiana hasła

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_ChangeLoginUser | Data1…n, 0x0 | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_ChangeLoginUser | Zmiana hasła | 0xb4 |
| Data1…n | jest dowolnym łańcuchem bajtów który będzie obowiązującym hasłem dostępu. | Dowolne z zakresu 0x01…0xff. Długość łańcucha może wynosić od 0 do 8 bajtów |
| 0x00 | Zero kończące string | 0x00 |

Jeżeli Data1=0x00 to czytnik nie będzie chroniony hasłem. W dowolnym momencie można ustalić nowe hasło tak aby czytnik był chroniony hasłem.

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_ChangeLoginUser+1 | | KodOperacji | CRC |

### 2.10.3 Wylogowanie z czytnika

Rozkaz ten dezaktualizuje podane ostatnio hasło.

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_LogoutUser | | CRC |

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_LogoutUser | Wylogowanie z czytnika | 0xd6 |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_LogoutUser +1 | | KodOperacji | CRC |

## 2.11 Zapis konfiguracji automatu

Rozkaz ten konfiguruje sposób pracy automatu odczytującego unikalny numer transpondera UID.

Ze względu na wysokie bezpieczeństwo danych jakie dają transpondery Mifare nie ma możliwości jednoczesnej pracy automatu odczytującego UID oraz komunikacji z transponderami poprzez łącze USB

Opisywany czytnik daje możliwość chwilowego zawieszania pracy automatu w przypadku wystąpienia poprawnej transmisji na łączu komunikacyjnym.

Jeżeli czytnik będzie pracował w trybie mieszanym, tzn.
-uruchomiony jest automat odczytów UID, oraz:
-urządzenie nadrzędne (komputer, sterownik) komunikuje się z czytnikiem albo za pomocą czytnika z transponderami
to:
konieczne jest odpowiednie skonfigurowanie czytnika tak aby w przypadku transmisji z czytnikiem lub z transponderem automat odczytów zawieszał swoją pracę.

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_SetAutoReaderConfig | ATrig, AOfflineTime, Aserial, AMode, ABuzz | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_SetAutoReaderConfig 0x58 | Zapis konfiguracji automatu | 0x58 |
| ATrig | Definiuje kiedy automat odczytów UID ma pracować | 0-automat wyłączony na stałe<br>1-automat załączony na stałe<br>2=załącza się automatycznie gdy brak transmisji na RS/USB przez czas dłuższy niż AOfflineTime<br>3= załącza się automatycznie gdy brak wywołań rozkazów komunikacji z transponderem przez czas dłuższy niż AOfflineTime |
| AOfflineTime | Czas braku transmisji na RS/USB<br>T= AofflineTime * [100 ms]<br>Brak transmisji może dotyczyć dowolnych rozkazów (Atrig=2), lub rozkazów komunikacji z transponderem (Atrig=3).<br><br>Rozkazy komunikacji z transponderem to:<br>C_TurnOnAntennaPower<br>C_Select | 0x00…0xff |

| | | | |
|---|---|---|---|
| | C_LoginWithDKB<br>C_LoginWithSKB)<br>C_ReadBlock<br>C_WriteBlock<br>C_CopyBlock<br>C_WritePage4B<br>C_ReadPage16B<br>C_IncrementValue<br>C_DecrementValue<br>C_WriteValue<br>C_ReadValue<br>C_Halt | | |
| ASerial | Automatyczne wysyłanie numeru transpondera UID po automatycznym odczytaniu go z transpondera | 0-nigdy<br>1-tylko za pierwszym przyłożeniem transpondera<br>2-wysyła wszystkie | |
| AMode | Wybór formatu wysyłanego numeru<br><br>8 bitów:<br>MSB<br>LSB<br><br>R L H C M E I A | R | Zarezerwowane, zawsze 0 |
| | | C=1<br>L=0 | numer kończy sie znakami końca wiersza CR+LF |
| | | C=1<br>L=1 | Numer kończy się znakiem końca wiersza LF |
| | | C=0<br>L=1 | Numer kończy się znakiem końca wiersza CR |
| | | M=1 | numer zaczyna się znakiem”M” |
| | | E=1 | informacja rozszerzona o ilość kart w polu oraz typ karty |
| | | I=1 | Numer w odwrotnej kolejności |
| | | A=1<br>H=0 | Numer wysyłany w formacie ASCII |
| | | A=0<br>H=0 | Numer wysyłany w formacie ramki Nertonix |
| | | A=0<br>H=1 | Numer wysyłany w formacie HEX |
| ABuzz | Automatyczne sygnalizowanie odczytu za pomocą buzzera po automatycznym odczytaniu UID’u z transpondera. | 0-nigdy<br>1-tylko za pierwszym przyłożeniem transpondera<br>2-sygnalizuje wszystkie | |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_ SetAutoReaderConfig +1 | | KodOperacji | CRC |

## 2.12 Odczyt konfiguracji automatu

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_ GetAutoReaderConfig | | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_GetAutoReaderConfig | Odczyt konfiguracji automatu | 0x5a |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_ GetAutoReaderConfig +1 | ATrig, AOfflineTime, ASerial, ABuzz | KodOperacji | CRC |

Gdzie:
Znaczenie parametrów odpowiedzi jest identyczne jak opisane wcześniej.

## 2.13 Konfiguracja interface'u szeregowego USB

### 2.13.1 Zapis konfiguracji interfejsu szeregowego

Rozkaz:

| | | | |
|---|---|---|---|
| | C_SetInterfaceConfig | Mode, Adr, Bodrate | |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_SetInterfaceConfig | zapis konfiguracji interfejsu szeregowego | 0x54 |
| Mode | | 0x01 |
| Adr | Adres na magistrali | 0x01…0xfe |
| baudrate | Prędkość danych na magistrali | 0x01=2400 b/s<br>0x02=4800 b/s<br>0x03=9600 b/s<br>0x04=19200 b/s<br>0x05=38400 b/s<br>0x06=57600 b/s<br>0x07=115200 b/s |

Odpowiedź:

| | | | | |
|---|---|---|---|---|
| | C_SetInterfaceConfig +1 | | KodOperacji | |

### 2.13.2 Odczyt konfiguracji interfejsu szeregowego

Rozkaz:

| | | | |
|---|---|---|---|
| | C_ GetInterfaceConfig | | |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_GetInterfaceConfig | odczyt konfiguracji interfejsu szeregowego | 0x56 |

Odpowiedź:

| | | | | |
|---|---|---|---|---|
| | C_ GetInterfaceConfig +1 | Mode, Adr, Bodrate | KodOperacji | |

Gdzie:
Znaczenie parametrów odpowiedzi jest identyczne jak opisane wcześniej.

## 2.14 Rozkazy pozostałe

### 2.14.1 Zdalny reset czytnika

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_Reset | | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_Reset | Zdalny reset czytnika | 0xd0 |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_Reset +1 | | KodOperacji | CRC |

### 2.14.2 Włączenie/wyłączenie funkcji emulacji klawiatury

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_Keyboard | [Param] | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_Keyboard | Włączenie/wyłączenie klawiatury | 0x04 |
| [Param] | Jeden bajt postaci:<br>MSB … LSB<br>0 0 0 0 H/D INV ENTER EN | H/D=1 – format heksadecymalny<br>H/D=0 – format dziesiętny<br>INV=1 – odwrócona kolejność<br>ENTER = 1 – symulacja wciśnięcia klawisza ENTER po wpisaniu ID<br>EN = 1 – klawiatura włączona |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_Keyboard +1 | Param | KodOperacji | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_Keyboard+1 | Włączenie/wyłączenie klawiatury | 0x05 |
| Param | Znaczenie pola takie jak dla rozkazu | |

### 2.14.3 Odczyt wersji oprogramowania czytnika

Ramka rozkazu:

| | | | |
|---|---|---|---|
| nagłówek | C_FirmwareVersion | | CRC |

Gdzie:

| Nazwa parametru | Opis parametru | Zakres wartości |
|---|---|---|
| C_FirmwareVersion | Odczyt wersji oprogramowania czytnika | 0xfe |

Ramka odpowiedzi:

| | | | | |
|---|---|---|---|---|
| nagłówek | C_FirmwareVersion+1 | Data1…..n | KodOperacji | CRC |

Gdzie

Data1…n jest ciągiem znaków zapisanych w postaci kodów ASCII.

## 2.15 Znaczenie kodów operacji w ramkach odpowiedzi

| Nazwa kodu operacji | Opis | wartość |
|---|---|---|
| OC_Error | błąd | 0x00 |
| OC_ParityError | błąd parzystości | 0x01 |
| OC_RangeError | Błąd zakresu parametru | 0x02 |
| OC_LengthError | Błąd ilości danych | 0x03 |
| OC_ParameterError | Błąd parametru | 0x04 |
| OC_Busy | Chwilowa zajętość wewnętrznych modułów | 0x05 |
| OC_BadCRC | Błąd komunikacji z kartą | 0x20 |
| OC_CommandUnknown | Nieznana komenda | 0x07 |
| OC_WrongPassword | Złe hasło lub ostatnie hasło uległo przeterminowaniu czyli miał miejsce automatyczny LogOut. | 0x09 |
| OC_NoCard | Brak transpondera | 0x0a |
| OC_BadFormat | Zły format danych. | 0x18 |
| OC_FrameError | Błąd transmisji. Może on świadczyć o istniejących zakłóceniach. | 0x19 |
| OC_NoAnswer | Brak odpowiedzi z transpondera | 0x1E |
| OC_TimeOut | Przekroczony czas operacji. Może on świadczyć o braku transpondera w polu czytnika | 0x16 |
| OC_Successful | Operacja zakończona poprawnie | 0xff |
| **Kody operacji związane z transponderami DESFIRE** | | |
| OC_DesNoChanges | Operacja Commit nie przyniosła zmian | 0x0c |
| OC_DesOutOfEEprom | Brak pamięci eeprom | 0x0e |
| OC_DesIllegalCommand | Niedozwolona komenda | 0x1c |
| OC_DesIntegrityError | Błąd CRC/ transmisji z kartą | 0x1e |
| OC_DesNoSuchKey | Nieprawidłowy numer klucza | 0x40 |
| OC_DesLengthError | Nieprawidłowa długość komendy | 0x7e |
| OC_DesPermisionDenied | Brak uprawnień do wykonania danej operacji | 0x9d |
| OC_DesParameterError | Błąd parametru komendy | 0x9e |
| OC_DesApplNotFound | Brak aplikacji o wybranych Aid | 0xa0 |
| OC_DesApplIntegrError | Błąd aplikacji, aplikacja zostaje zablokowana | 0xa1 |
| OC_DesAuthError | Błąd autoryzacji / niepoprawny klucz | 0xae |
| OC_DesBoundaryError | Zapis/odczyt z rekordu wykroczył poza wielkość | 0xbe |
| OC_DesPICCIntegError | Wewnętrzny błąd transpondera, zostaje zablokowany | 0xc1 |
| OC_DesCountError | Przekroczony limit 28 aplikacji | 0xce |
| OC_DesDuplicateError | Aplikacja/Plik o tym identyfikatorze już istnieje | 0xde |
| OC_DesEepromError | Błąd podczas zapisu/odczytu do pamięci EEPROM | 0xee |
| OC_DesFileNotFound | Plik o tym identyfikatorze nie istnieje | 0xf0 |
| OC_DesFileIntegrError | Nieodwracalny błąd pliku, plik zostaje zablokowany | 0xf1 |

# 3 Emulacja klawiatury

Urządzenie PAC-PUx może emulować klawiaturę USB (HID). Podczas emulowania klawiatury, każde odczytanie ID transpondera poprzez mechanizm AutoReader'a powoduje symulację wpisania jego ID. Przykładowy format wysłanego ID w zależności od konfiguracji został przedstawiony poniżej:

| ID transpondera: 0x1C34AB1F55 | | | | |
|---|---|---|---|---|
| Konfiguracja | | | | Format |
| **H/D** | **INV** | **ENTER** | **Caps Lock** | |
| 0 | 0 | 0 | X | 121142714197 |
| 0 | 0 | 1 | X | 121142714197 <ENTER> |
| 0 | 1 | 0 | X | 369126749212 |
| 0 | 1 | 1 | X | 369126749212 <ENTER> |
| 1 | 0 | 0 | Off | 1c34ab1f55 |
| 1 | 0 | 0 | On | 1C34AB1F55 |
| 1 | 0 | 1 | Off | 1c34ab1f55 <ENTER> |
| 1 | 0 | 1 | On | 1C34AB1F55 <ENTER> |
| 1 | 1 | 0 | Off | 551fab341c |
| 1 | 1 | 0 | On | 551FAB341C |
| 1 | 1 | 1 | Off | 551fab341c <ENTER> |
| 1 | 1 | 1 | On | 551FAB341C <ENTER> |

# 4 Powrót do ustawień fabrycznych

Aby powrócić do ustawień fabrycznych należy na czas ok. 5 sekund przycisnąć, znajdujący się w małym otworze na spodzie obudowy.
Podczas powrotu do ustawień fabrycznych ustawiane są na stałe następujące parametry czytnika:

| Nazwa parametru lub funkcjonalność | Wartość lub ustawienie |
|---|---|
| Hasło dostępu | Brak hasła |
| Port 0 – LED1 | sygnalizacja odczytu karty, sterowanie portem szeregowym |
| Port 1 – LED2 | sterowanie portem szeregowym |
| Port 3 – BUZZER | sygnalizacja odczytu karty, sterowanie portem szeregowym |
| Autoreader | włączony |
| Klawiatura | wyłączona |

# 5 Przykład pracy z transponderemi

## 5.1 Przykład pracy z transponderem S50, S70

Po poprawnym podłączeniu czytnika i nawiązaniu obustronnej komunikacji pomiędzy nim a komputerem nadrzędnym można przystąpić do operacji odczytu i zapisu pamięci transpondera. Poniższe operacje zakładają, że czytnik posiada ustawienia fabryczne oraz, że użyta karta S50 posiada ustawienia fabryczne czyli pełne prawa dostępu i oba klucze 0xff ff ff ff ff ff.

Ponieważ podczas ręcznych prób czas pomiędzy kolejnymi rozkazami wysyłanymi po RS jest stosunkowo duży i osiąga od kilku sekund do kilku minut to należy wyłączyć wewnętrzny automat odczytów UID.
Należy to zrobić za pomocą rozkazu :
**SetAutoReaderConfig 0x00, 0x00, 0x00, 0x00, 0x00**

Aby dokonać odczytu transpondera, najpierw należy załadować klucz do pamięci kluczy. Załadujmy więc klucz do SKB za pomocą
**C_LoadKeyToSKB 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00**

Załączamy pole.
**TurnOnAntennaPower 0x01**

Przykładamy transponder do czytnika,
Selekcjonujemy transponder
**C_Select 0x00**

Logujemy się np. do sektora 3.
**C_LoginWithSKB 0x03, 0xAA, 0x00**

Odczytajmy zawartość 2-go bloku w 3-cim sektorze.
**C_ReadBlock 0x02**

O ile wszystkie Kody Operacji w ramkach odpowiedzi były OC_Successful to otrzymane wartości są danymi odczytanymi z bloku.

## 5.2 Przykład pracy z transponderami DESFire

Po poprawnym podłączeniu czytnika i nawiązaniu obustronnej komunikacji pomiędzy nim a komputerem nadrzędnym można przystąpić do operacji odczytu i zapisu pamięci transpondera.
Poniższe operacje zakładają, że czytnik posiada ustawienia fabryczne oraz, że użyta karta Desfire posiada ustawienia fabryczne czyli pełne prawa dostępu, a klucz PICC Master key ma wartość 0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00.

Wynikiem tego przykładu jest utworzenie nowej aplikacji, zmiana standardowego klucza aplikacji, utworzenie pliku z danymi, zapisanie a następnie odczyt danych z pliku.

Ponieważ podczas ręcznych prób czas pomiędzy kolejnymi rozkazami wysyłanymi po RS jest stosunkowo duży i osiąga od kilku sekund do kilku minut to należy wyłączyć wewnętrzny automat odczytów UID.
Należy to zrobić za pomocą rozkazu :

1. **SetAutoReaderConfig 0x00, 0x00, 0x00, 0x00.**

Aby dokonać odczytu transpondera, najpierw należy załadować klucze do pamięci kluczy.
Ładujemy więc standardowy klucz transponderów desfire na pozycję np.„3” pamięci czytnika, a na pozycję 4 ładujemy sobie własny klucz, który nadamy nowej aplikacji:

2. **C_DesSaveKey 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00**

3. **C_DesSaveKey 0x04, 0x01, 0x02, 0x03, 0x04, 0x0a, 0x0b, 0x0c, 0x0d**

Załączamy pole.

4. **C_TurnOnAntennaPower 0x01**

Przykładamy transponder do czytnika, Selekcjonujemy transponder

5. **C_Select 0x00**

Inicjujemy protokół wymiany danych ISO, z numerem logicznym transpondera 0

6. **C_DesInitProtocol 0x00**

Dokonujemy autoryzacji z użyciem klucza „0” czyli PICC Master key, klucz ten zapisany jest w pamięci czytnika pod indeksem „3”

7. **C_DesAuth 0x00,0x03**

Tworzymy aplikację o numerze identyfikacyjnym np. 0x30, 0x10, 0x55, o domyślnych ustawieniach klucza ApplicationMasterKey, z rezerwacją miejsca na 4 klucze

8. **C_DesCreateApp 0x30,0x10,0x55,0x0F,0x04**

Zmieniamy domyślny, nowo utworzony klucz ApplicationMasterKey na ten, który mamy zapisany w czytniku na pozycji 4. W związku z tym selekcjonujemy nową aplikację:

9. **C_DesSelectApp 0x30,0x10,0x55**

Logujemy się do aplikacji z użyciem klucza Application Master Key, a następnie zmieniamy go po czym ponownie logujemy z użyciem nowego klucza

**10. C_DesAuth 0x00,0x03**
**11. C_DesChangeKey 0x00,0x04**
**12. C_DesAuth 0x00,0x04**

Tworzymy standardowy plik z danymi, z pełnymi prawami dostępu dla Application Master Key, oraz prawami odczytu dla klucza „3”. Plik będzie miał indeks „2”, nieszyfrowaną wymianę danych oraz wielkość 1500 bajtów

**13. C_DesCreateSTDataFile 0x02,0x00,0x30,0x00,0xDC,0x05,0x00**

Dokonujemy teraz zapisu danych do utworzonego właśnie pliku od pozycji 0

**14. C_DesWriteData 0x02,0x00,0x00,0x00, $TuSaNaszeDaneDoZapisu**

Odczytujemy 21 bajtów właśnie zapisanych danych

**15. C_DesReadData 0x02,0x00,0x00,0x00, 0x15,0x00,0x00**

## 5.3 Przykład pracy z transponderami Mifare Plus

Po poprawnym podłączeniu czytnika i nawiązaniu obustronnej komunikacji pomiędzy nim a komputerem nadrzędnym, można przystąpić do operacji odczytu i zapisu pamięci transpondera.

Poniższe operacje zakładają, że czytnik posiada ustawienia fabryczne oraz, że użyta niezainicjowana, nowa karta Mifare Plus S 2kB/4kB.

Poniższy przykład prezentuje:

- załadowanie kluczy AES do pamięci czytnika,
- załadowanie podstawowych kluczy AES do pamięci transpondera,
- przejście do poziomu SL1,
- uwierzytelnienie AES na poziomie SL1,
- zapis bloku na poziomie SL1,
- odczyt bloku na poziomie SL1,
- przejście do poziomu SL3,
- logowanie AES do sektora ,
- zapis bloku metodą MAC on command, MAC on response (jedyną dostępną dla Mifare Plus S),
- odczyt bloku metodą MAC on command, MAC on response (jedyną dostępną dla Mifare Plus S)

Przykłady można zrealizować za pomocą darmowych aplikacji **Framer4** lub **MFPlus Tool**.

Ponieważ podczas ręcznych prób czas pomiędzy kolejnymi rozkazami wysyłanymi po RS jest stosunkowo duży i osiąga od kilku sekund do kilku minut to należy wyłączyć wewnętrzny automat odczytów UID.
Należy to zrobić za pomocą rozkazu :

**SetAutoReaderConfig 0x00, 0x00, 0x00, 0x00, 0x00**

Pierwszym etapem jest załadowanie kluczy do pamięci czytnika. Będą one następnie wykorzystane przy inicjalizacji karty, zmiany poziomu SL oraz logowaniu do poszczególnych sektorów karty.

**C_DesSaveKey 0x01, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,**
**0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF**

**C_DesSaveKey 0x03, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88,**
**0x99, 0x00, 0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF**

**C_DesSaveKey 0x04, 0x01, 0x02, 0x03, 0x04, 0x0a, 0x0b, 0x0c, 0x0d,**
**0x0e, 0x0f, 0x10, 0x12, 0x14, 0x16, 0x18, 0x20**

oraz domyślny klucz Mifare Classic na pozycję 0 pamięci czytnika

**C_LoadKeyToSKB 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00**

Załączamy pole.

**TurnOnAntennaPower 0x01**

Przykładamy transponder do czytnika

Selekcjonujemy Transponder

**C_Select 0x00**

Zapisujemy klucz ‘Card Master Key’ (taki jak uprzednio wprowadzony pod indeksem 0x03)

**C_ MfPlusCMD 0xA8 0x90 0x00 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0x00, 0xAA, 0xBB, 0xCC, 0xDD, 0xEE, 0xFF**

Zapisujemy klucz ‘SL1 Auth Key’ (taki jak uprzednio wprowadzony pod indeksem 0x04)

**C_ MfPlusCMD 0xA8 0x90 0x04 0x01, 0x02, 0x03, 0x04, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f, 0x10, 0x12, 0x14, 0x16, 0x18, 0x20**

Zapisujemy klucz ‘Level 3 Switch Auth Key’ (taki jak uprzednio wprowadzony pod indeksem 0x04)

**C_ MfPlusCMD 0xA8 0x90 0x03 0x01, 0x02, 0x03, 0x04, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f, 0x10, 0x12, 0x14, 0x16, 0x18, 0x20**

Zapisujemy klucz AES typu A dla sektora 0x01(taki jak uprzednio wprowadzony pod indeksem 0x03)

**C_ MfPlusCMD 0xA8 0x40 0x02 0x01, 0x02, 0x03, 0x04, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f, 0x10, 0x12, 0x14, 0x16, 0x18, 0x20**

Przechodzimy do poziomu SL1, wykonując komendę COMMIT PERSO

**C_ MfPlusCMD 0xAA**

W tym momencie karta musi być zresetowana, zdejmujemy ją na moment z pola działania czytnika, a następnie selekcjonujemy ją ponownie

**C_Select 0x00**

Wykonujemy autoryzacje AES kluczem o indeksie 4

**C_ MfPlusCMD 0x10 0x04**

Logujemy się np. do sektora 3 używając klucza A.

**C_LoginWithSKB 0x03, 0xAA, 0x00**

Zapisujemy zawartość 2-go bloku w 3-cim sektorze przykładowymi wartościami.

**C_WriteBlock 0x02 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 0x99 0xaa 0xbb 0xcc**

**0xdd 0xee 0xff 0x00**

Odczytajmy zawartość 2-go bloku w 3-cim sektorze.

**C_ReadBlock 0x02**

W tym momencie (przed przejściem do ISO14443-4) karta musi być zresetowana, zdejmujemy ją na moment z pola działania czytnika, a następnie selekcjonujemy ją ponownie
**C_Select 0x00**

Od tego momentu transmisja z transponderem odbywać się będzie wg ISO14443-4, konieczne jest zainicjowanie tego trybu

**C_Init_ISO14443-4 0x00**

Przechodzimy do poziomu SL3, wykonując uwierzytelnianie kluczem ‘SL3 Auth Key’, który mamy zapisamy pod indeksem 4

**C_ MfPlusCMD 0x70 0x90 0x03 0x0x04**

W tym momencie, po przejściu do SL3, karta musi być zresetowana, zdejmujemy ją na moment z pola działania czytnika, a następnie selekcjonujemy ją ponownie

**C_Select 0x00**
**C_Init_ISO14443-4 0x00**

Logujemy się do sektora 1 używając klucza A, który uprzednio został zapisany pod indeksem 3.
**C_ MfPlusCMD 0x1A, 0x01, 0xAA, 0x03**

Zapisujemy blok 2 sektora 1 przykładowymi wartościami

**C_ MfPlusCMD 0xA3 0x02 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 0x99 0xaa 0xbb 0xcc 0xdd 0xee 0xff 0x00**

Odczytujemy blok 2 sektora 1

**C_ MfPlusCMD 0x33 0x02**

Najnowsze wiadomości  dotyczące produktów firmy NETRONIX
http://www.netronix.pl/