



ND66-v13 APB D251120

# DOKUMENTACJA TECHNICZNA MW-R7B / MW-R7G



MW-R7G

MW-R7B

Wersja dokumentacji: MW-R7x-MAN-V13  
Data: 25.11.2020  
Obowiązuje od wersji firmware: MW-R7x-v3.6.A1.6

## SPIS TREŚCI:

<b>1</b>	<b>Wprowadzenie</b>	<b>5</b>
<b>2</b>	<b>Dane techniczne</b>	<b>6</b>
<b>3</b>	<b>Wyprowadzenia</b>	<b>7</b>
<b>4</b>	<b>Wejścia / wyjścia</b>	<b>8</b>
4.1	<i>Wejścia fizyczne</i>	8
4.1.1	Parametry portów wejściowych PinIN	8
4.2	<i>Wyjścia fizyczne</i>	8
4.2.1	Parametry portu wyjściowego PINOUT	8
4.3	<i>Dioda RGB</i>	8
4.4	<i>Źródła sygnałów sterujących wyjściami</i>	8
4.4.1	Źródło „0” oraz źródło „1”	9
4.4.2	Źródło przycisk	9
4.4.3	Źródło każda karta	9
4.4.4	Źródło RSx	9
4.4.5	Źródło PinINx	9
4.4.6	Źródło SIG_Ax	10
4.4.7	Źródło SIG_Bx	10
4.4.8	Źródło SIG_Cx	10
<b>5</b>	<b>Wymiary</b>	<b>11</b>
<b>6</b>	<b>Interfejs</b>	<b>12</b>
6.1	<i>Interfejs 1-WIRE</i>	12
6.2	<i>Interfejs WIEGAND</i>	12
6.3	<i>Interfejs RS232(TTL) / RS485 / CAN</i>	12
6.3.1	Protokół komunikacji dla interfejsu RS232(TTL) / 485	13
6.3.2	Protokół komunikacji dla interfejsu CAN	13
6.3.3	Poziomy sygnałów dla szeregowej transmisji RS232 (TTL)	13
<b>7</b>	<b>Komendy dostępne dla interfejsu RS232(TTL) / RS485 / CAN – protokół NETRONIX</b>	<b>15</b>
7.1	<i>Konfiguracja interfejsu szeregowego</i>	15
7.1.1	Zapis konfiguracji interfejsu szeregowego	15
7.1.2	Odczyt konfiguracji interfejsu szeregowego	15
7.2	<i>Rozkazy komunikacji z transponderami</i>	16
7.2.1	Zarządzanie kluczami	16
7.2.1.1	Zapis klucza MIFARE Classic do dynamicznej pamięci kluczy	17
7.2.1.2	Zapis klucza MIFARE Classic do statycznej pamięci kluczy	17
7.2.1.3	Zapis klucza AES / 3DES do statycznej pamięci kluczy	17
7.2.2	Rozkazy wspólne do komunikacji z transponderami	17
7.2.2.1	Załączanie i wyłączenie pola czytnika	17
7.2.2.2	Wyselekcjonowanie jednego transpondera z wielu	18
7.2.2.3	Uśpienie transpondera będącego w polu	18
7.2.3	Rozkazy do komunikacji z transponderami Mifare Classic	18
7.2.3.1	Logowanie do sektora transpondera za pomocą Dynamicznego Klucza	18
7.2.3.2	Logowanie do sektora transpondera za pomocą Statycznego Bufora Kluczy	19
7.2.3.3	Odczyt zawartości bloku transpondera	20
7.2.3.4	Zapis zawartości bloku transpondera	20
7.2.3.5	Kopiowanie zawartości bloku transpondera do innego bloku	20
7.2.3.6	Zapis wartości do bloku transpondera	20
7.2.3.7	Odczyt wartości z bloku transpondera	21
7.2.3.8	Zwiększenie wartości zawartej w bloku transpondera	21
7.2.3.9	Zmniejszanie wartości zawartej w bloku transpondera	21
7.2.4	Rozkazy do komunikacji z transponderami Mifare Ultralight	22
7.2.4.1	Zapis zawartości strony w Mifare UL	22
7.2.4.2	Odczyt zawartości stron w Mifare UL	22
7.2.4.3	Uwierzytelnienie dla transpondera Ultralight C	22
7.2.5	Rozkazy do komunikacji z transponderami Mifare Plus	22
7.2.5.1	Rozkazy poziomu SL0	22
7.2.5.2	Rozkazy poziomu SL1	23
7.2.5.3	Przejdźcie na wyższy poziom SL / sprawdzenie oryginalności transpondera	23
7.2.5.4	Rozkazy poziomu SL3	24
7.2.5.5	Czasy trwania operacji dla Mifare Plus	25

7.2.6	Obsługa transponderów DESFire, DESFire EV1 .....	25
7.2.6.1	Autoryzacja, logowanie do aktualnie wyselekcjonowanej aplikacji .....	25
7.2.6.2	Zmiana ustawień klucza Master aktualnie wybranej aplikacji .....	26
7.2.6.3	Zmiana klucza .....	26
7.2.6.4	Tworzenie aplikacji .....	26
7.2.6.5	Usuwanie aplikacji .....	27
7.2.6.6	Pobieranie listy aplikacji .....	27
7.2.6.7	Wybór aplikacji .....	28
7.2.6.8	Formatowanie transpondera .....	28
7.2.6.9	Inicjalizacja protokołu transmisji z transponderami DESFire .....	28
7.2.6.10	Pobieranie listy plików aktualnie wybranej aplikacji .....	28
7.2.6.11	Pobieranie właściwości pliku .....	28
7.2.6.12	Tworzenie plików typu <i>Standard Data Files</i> .....	29
7.2.6.13	Tworzenie plików typu <i>Backup Data Files</i> .....	30
7.2.6.14	Tworzenie plików typu <i>Linear/Cyclic Record Files</i> .....	30
7.2.6.15	Usuwanie pliku .....	30
7.2.6.16	Zmiana ustawień pliku .....	30
7.2.6.17	Odczyt danych z pliku typu <i>Std/Back Data File</i> .....	31
7.2.6.18	Zapis danych do pliku typu <i>Std/Back Data File</i> .....	31
7.2.6.19	Zapis rekordu do pliku typu <i>Record Data File</i> .....	31
7.2.6.20	Odczyt rekordu z pliku typu <i>Record Data File</i> .....	32
7.2.6.21	Czyszczenie plików typu <i>Record Data File</i> .....	32
7.2.6.22	Komenda potwierdzająca - <i>DesCommit</i> .....	32
7.2.6.23	Deselekcja transpondera .....	33
7.2.7	Transmisja danych I-Block protokołu T=CL ISO14443-4 .....	33
7.2.8	Obsługa transponderów rodziny I-CODE SLI .....	33
7.2.8.1	Odczyt numeru ID transpondera I-CODE SLI .....	33
7.2.8.2	Odczyt strony transpondera SLI .....	33
7.2.8.3	Zapis zawartości strony w SLI .....	33
7.2.9	Mifare Application Directory - MAD .....	34
7.2.9.1	Formatowanie karty MAD .....	34
7.2.9.2	Dodanie aplikacji do katalogu MAD .....	34
7.2.9.3	Wyszukanie sektora dla danej aplikacji .....	35
7.2.9.4	Wyszukanie kolejnego sektora aplikacji .....	35
7.3	<i>Źródła oraz wejścia i wyjścia elektryczne</i> .....	35
7.3.1	Zapis stanu źródła RSx .....	35
7.3.2	Odczyt stanu źródła .....	36
7.3.3	Zapis konfiguracji portu .....	36
7.3.4	Odczyt konfiguracji portu .....	37
7.3.5	Konfiguracja bloku SIG_A .....	37
7.3.6	Konfiguracja bloku SIG_B .....	38
7.3.7	Konfiguracja bloku SIG_C .....	38
7.3.8	Konfiguracja kolorów .....	39
7.4	<i>Hasło dostępu</i> .....	39
7.4.1	Logowanie do czytnika .....	39
7.4.2	Zmiana hasła .....	39
7.4.3	Wylogowanie z czytnika .....	40
7.5	<i>Mechanizm AutoReader</i> .....	40
7.5.1	Zapis konfiguracji automatu .....	40
7.5.2	Odczyt konfiguracji automatu .....	41
7.6	<i>Obsługa ID zapisanych w pamięci transpondera</i> .....	42
7.6.1	Zapis/odczyt konfiguracji .....	42
7.7	<i>Rozkazy pozostałe</i> .....	42
7.7.1	Zdalny reset czytnika .....	42
7.7.2	Odczyt wersji oprogramowania czytnika .....	42
7.8	<i>Znaczenie kodów operacji w ramach odpowiedzi</i> .....	43
<b>8</b>	<b>Protokół MODBUS RTU</b> .....	<b>44</b>
8.1	<i>Obsługiwane funkcje protokołu MODBUS</i> .....	44
8.2	<i>Adresy MODBUS</i> .....	44
8.2.1	Adresy do odczytu ID karty .....	44

8.2.2	Adresy do odczytu/zapisu konfiguracji autoreadera.....	44
8.2.3	Adresy do odczytu/zapisu konfiguracji interfejsów RS232 / RS48 .....	44
8.2.4	Adresy do odczytu/zapisu konfiguracji bloków SigA .....	44
8.2.5	Adresy do odczytu/zapisu konfiguracji bloków SigB .....	45
8.2.6	Adresy do odczytu/zapisu konfiguracji bloków SigC .....	45
8.2.7	Adresy do odczytu/zapisu konfiguracji LED.....	45
8.2.8	Adresy do odczytu/zapisu konfiguracji I/O .....	46
8.2.9	Adresy do odczytu/zapisu konfiguracji mechanizmu UserID .....	46
8.2.10	Adresy do zapisu wartości źródeł RSx .....	46
8.2.11	Adresy do odczytu wartości źródeł .....	46
8.3	<i>Enkapsulacja protokołu Netronix w protokole MODBUS RTU.....</i>	47
8.3.1	Schemat postępowania .....	47
8.3.2	Przykład użycia – odczyt wersji oprogramowania .....	47
<b>9</b>	<b>Powrót do ustawień fabrycznych.....</b>	<b>49</b>
<b>10</b>	<b>Bootloader - zmiana firmware urządzenia .....</b>	<b>50</b>
10.1	<i>Aktualizacja z wykorzystaniem aplikacji NEFIR 1.9.1 .....</i>	<i>50</i>
10.2	<i>Aktualizacja z wykorzystaniem aplikacji NEFIR 3 .....</i>	<i>51</i>

## 1 WPROWADZENIE

MW-R7x jest ściennym czytnikiem kart RFID działających na częstotliwości znamionowej 13,56MHz.

Posiada on następującą cechy:

- Obsługa transponderów: MIFARE® Classic, Plus, Ultralight C, DESFire, I-CODE SLI, iClass (tylko CSN)
- Interfejsy:
  - RS232
  - RS485
  - 1-WIRE
  - WIEGAND
  - CAN
- Wbudowany buzzer,
- Wbudowana dioda LED RGB dowolnego przeznaczenia
- Przycisk frontowy
- Dostępny w kolorach:
  - czarny (MW-R7B)
  - beżowy (MW-R7G)

**2 DANE TECHNICZNE**

Typ transpondera	Odczyt numeru ID	Pełny zapis i odczyt bloków pamięci
MIFARE® Classic S50	TAK	TAK
MIFARE® Classic S70	TAK	TAK
MIFARE® Plus	TAK	SL0, SL1, SL3
Ultra Light	TAK	TAK
Ultra Light C	TAK	TAK
DESFire	TAK	TAK
I-CODE SLI	TAK	TAK
iClass	TAK (nr CSN)	NIE

Parametry czytnika MW-R7x	
Napięcie zasilania	8-15V
Maksymalny prąd zasilania	120 mA
Znamionowa częstotliwość RF pracy modułu	13,56MHz
Odległość odczytu transponderów	do 8 cm
Wymiary (szer.* dł. * wys.)	44x83x14 mm
Klasa szczelności IP	IP54
Przycisk frontowy	Tak - pojemnościowy
Interfejsy	RS-232 (TTL) RS-485 Wiegand 1-WIRE CAN
Wejścia / wyjście	Antykolizja (Wej/Wyj) PinOUT (Wyj) PinIN0 (Wej) PinIN1 (Wej)

### 3 WYPROWADZENIA

Przewód	Nazwa	Funkcja
<b>Czerwony</b>	VCC	Zasilanie (+)
<b>Niebieski</b>	GND	Zasilanie (-)
<b>Biały</b>	Antykolizja	Wyprowadzenie służące do połączenia ze sobą czujników pracujących blisko siebie.
<b>Brązowy</b>	PinOUT	Wyjście dowolnego przeznaczenia
<b>Zielony</b>	PinINTERFACE1	Linia interfejsu szeregowego (RS232_TX, RS485_B, CAN_H, WIEGAND0, 1WIRE)
<b>Żółty</b>	PinINTERFACE2	Linia interfejsu szeregowego (RS232_RX, RS485_A, CAN_L, WIEGAND1)
<b>Szary</b>	PinIN0	Wejście dowolnego przeznaczenia
<b>Różowy</b>	PinIN1	Wejście dowolnego przeznaczenia

## 4 WEJŚCIA / WYJŚCIA

### 4.1 WEJŚCIA FIZYCZNE

Czytnik MW-R7x posiada trzy wejścia fizyczne:

1. PinIN0
2. PinIN1
3. Przycisk

#### 4.1.1 PARAMETRY PORTÓW WEJŚCIOWYCH PININ

Poziom logiczny „0”	0 ...0.2V
Poziom logiczny „1”	3.75...25V

Wejście są podciągane przez rezystor 10kOhm do +5V.

### 4.2 WYJŚCIA FIZYCZNE

Czytnik MW-R7x posiada sześć wyjść fizycznych:

1. Kolor0 (dioda RGB)
2. Kolor1 (dioda RGB)
3. Kolor2 (dioda RGB)
4. Kolor3 (dioda RGB)
5. Buzzer
6. PinOUT

#### UWAGA:

*Stan aktywny wyjścia buzzer powoduje zablokowanie mechanizmu odczytów transponderów.*

#### 4.2.1 PARAMETRY PORTU WYJSCIOWEGO PINOUT

Typ wyjścia	Otwarty kolektor
Maksymalne napięcie w stanie wysokiej impedancji	40V
Maksymalny prąd	400mA

### 4.3 DIODA RGB

Czytnik MW-R7x przy użyciu diody RGB może wyświetlić 4 kolory: biały, czerwony, zielony oraz niebieski. Kody kolorów zostały zebrane w tabeli poniżej:

Tabela 4.1 Tabela kodów kolorów

Kod koloru	Kolor
0	Czerwony
1	Zielony
2	Niebieski
3	Biały

Przypisaną konkretnego koloru do wyjścia KolorX zostało opisane w podrozdziale „Konfiguracja kolorów”.

### 4.4 ŹRÓDŁA SYGNAŁÓW STERUJĄCYCH WYJŚCIAMI

Czytnik MW-R7x ma 18 źródeł sygnałów logicznych. Sygnały te mogą być wykorzystywane do sterowania wyjściami. W tabeli poniżej znajduje się zestawienie wszystkich źródeł oraz wartości sygnałów przez nie generowane.

Tabela 4.2 Źródła sygnałów

ID	Nazwa	Opis
0	„0”	Źródło sygnału o wartości 0
1	„1”	Źródło sygnału o wartości 1
2	Przycisk	Źródło odzwierciedlające stan przycisku frontowego. Ma wartość 1, gdy przycisk jest wciśnięty i wartość 0 w przeciwnym przypadku.



3	Każda karta	Źródło odzwierciedlające informację o obecności karty w polu czytnika. Ma wartość 1, gdy karta jest w polu i wartość 0 w przeciwnym przypadku.
4	RS_0	Źródła sterowane poprzez interfejs szeregowy RS232/RS485. Patrz komenda <i>C_WriteOutput</i>
5	RS_1	
6	RS_2	
7	RS_3	
8	PinIN0	
9	PinIN1	
10	SigA0	Źródła sterowane poprzez wyjścia bloków SIG_Ax
11	SigA1	
12	SigA2	
13	SigA3	
14	SigB0	Źródła sterowane poprzez wyjścia bloków SIG_Bx
15	SigB1	
16	SigB2	
17	SigB3	
18	SigC0	Źródła sterowane poprzez wyjścia bloków SIG_Cx
19	SigC1	
20	SigC2	
21	SigC3	

#### 4.4.1 ŹRÓDŁO „0” ORAZ ŹRÓDŁO „1”

Źródło sygnału „0” ma zawsze wartość równą 0, natomiast źródło sygnału „1” ma wartość równą 1.

#### 4.4.2 ŹRÓDŁO PRZYCIISK

Źródło odzwierciedlające stan przycisku frontowego. Ma wartość 1, gdy przycisk jest wciśnięty i wartość 0 w przeciwnym przypadku.

##### UWAGA:

W przypadku, gdy przycisk będzie wciśnięty dłużej niż 3 minuty, nastąpi rekalkibracja przycisku i wyzerowanie wartości źródła przycisk.

#### 4.4.3 ŹRÓDŁO KAŻDA KARTA

Źródło odzwierciedlające informację o obecności karty w polu czytnika. Ma wartość 1, gdy karta jest w polu i wartość 0 w przeciwnym przypadku.

#### 4.4.4 ŹRÓDŁO RSX

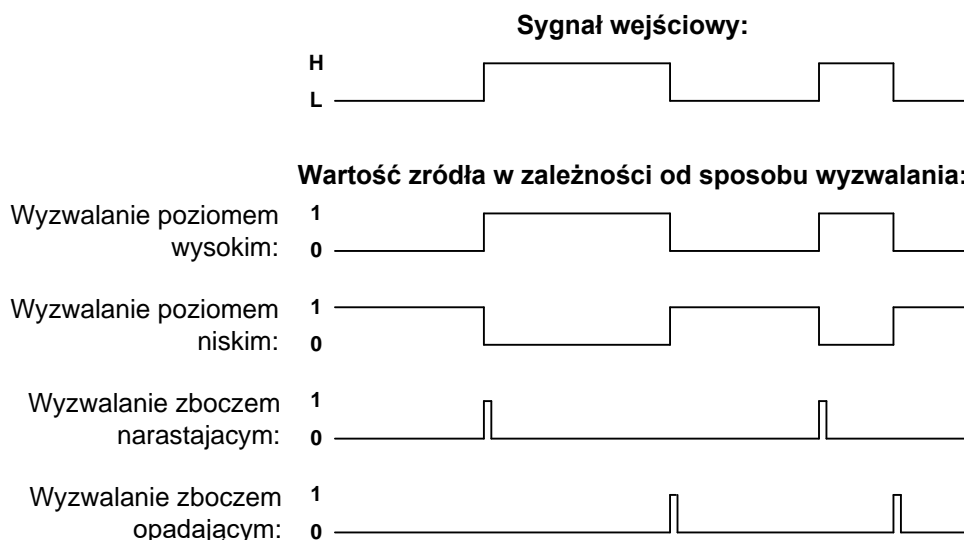
Źródła sterowane poprzez interfejs szeregowy RS232/RS485. Źródło umożliwia:

- Ustawienie wartości 0
- Ustawienie wartości 1
- Ustawienie wartości 1 na określony czas, po którym źródło automatycznie zmieni stan na 0.

Patrz komenda Zapis stanu źródła RSx.

#### 4.4.5 ŹRÓDŁO PININX

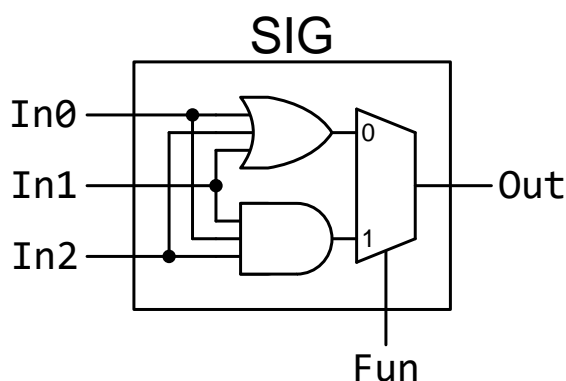
Źródła PinINx są sterowane poprzez fizyczne wejścia. W zależności od konfiguracji, wartość źródła ma wartość:



Konfiguracji sposobu wyzwalania dokonuje się przy użyciu komendy Zapis konfiguracji portu.

#### 4.4.6 ŹRÓDŁO SIG\_AX

Czytnik MW-R7x posiada 4 wirtualne bloki SIG\_A, które pozwalają na wykonywanie operacji logicznych na sygnałach. Każdy z bloków ma 3 wyjścia sygnałowe, jedno wejście wyboru funkcji oraz jedno wyjście. Do wejść sygnałowych bloków można podłączyć dowolne źródło sygnału. Na wyjściu bloku w zależności od wybranej funkcji *Fun* będzie znajdować się suma logiczna lub iloczyn logiczny sygnałów wejściowych. Konfiguracji bloków SIG dokonuje się przy użyciu komendy Konfiguracja bloku SIG\_A.

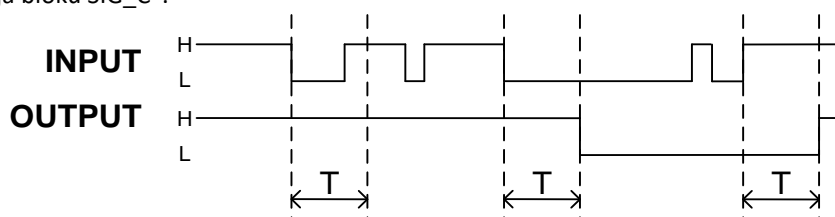


#### 4.4.7 ŹRÓDŁO SIG\_BX

Czytnik MW-R7x posiada 4 wirtualne bloki SIG\_B, które pozwalają na wykonywanie operacji logicznych na sygnałach. Konfiguracja bloków SIG\_B została opisana w podrozdziale „Konfiguracja bloku SIG\_B”.

#### 4.4.8 ŹRÓDŁO SIG\_CX

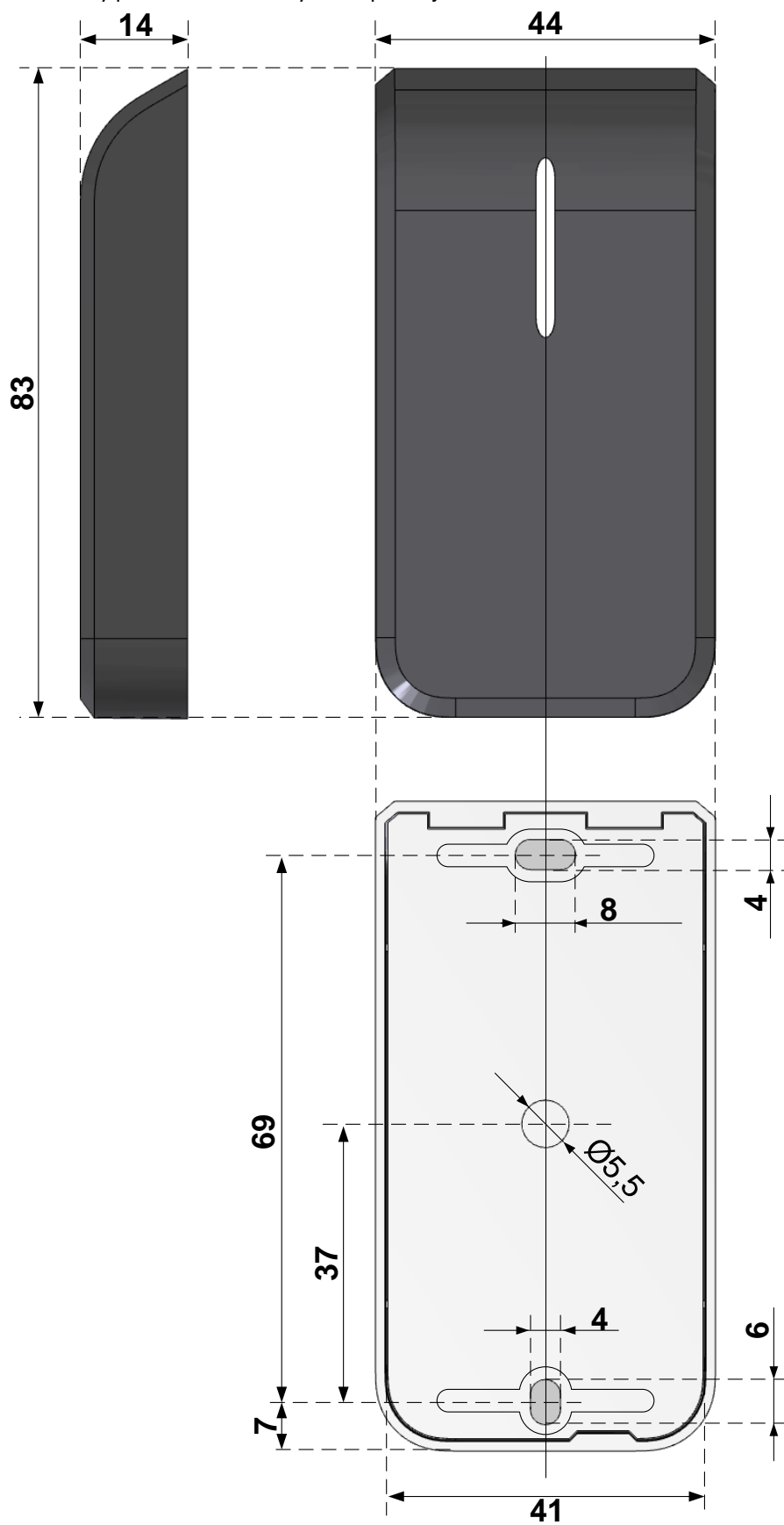
Czytnik MW-R7x posiada 4 wirtualne bloki SIG\_C, które pozwalają na wykonywanie filtrowania sygnałów logicznych. Stan na wyjściu bloku SIG\_C zmieni się na taki sam jak na wejściu, jeśli stan na wejściu będzie stały przez czas zdefiniowany przez parametr *Time*. Konfiguracja bloków SIG\_C została opisana w podrozdziale „Konfiguracja bloku SIG\_C”.



4.1 Przykładowy przebieg wejściowy i wyjściowy dla bloku SIG\_C.

## 5 WYMIARY

Wymiary czytnika zostały przedstawione na rysunku poniżej:



Długość przewodu: 30cm

## 6 INTERFEJS

Czytnik MW-R7x obsługuje 5 interfejsów szeregowych:

- RS232
- RS485
- 1-WIRE
- Wiegand
- CAN

Interfejsy RS-232 oraz RS-485/CAN cały czas nasłuchują w oczekiwaniu na komendę.

Automat AutoReader wysyła odczytanie ID poprzez interfejs wybrany w konfiguracji AutoReadera.

### 6.1 INTERFEJS 1-WIRE

Po skonfigurowaniu urządzenia do pracy w trybie 1-WIRE, czytnik emuluje zachowanie „pastylki” Dallas serii DS1990. Tak długo jak karta będzie w polu, tak długo czytnik będzie wystawiał unikalny numer na magistrali 1-WIRE. Czytnik obsługuje komendy READ\_ROM oraz SEARCH\_ROM. Format wysłanego ID ma postać:

Kod rodziny	ID transpondera					Adres	CRC
ConfFC	ID0	ID1	ID2	ID3	ID4	ConfAdr	xx

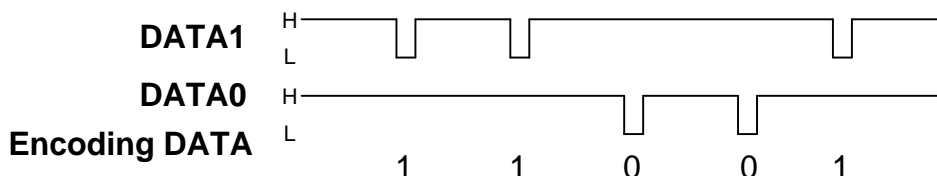
W przypadku konieczności zmiany parametru „Adres” lub „Kod rodziny” należy zmienić konfigurację interfejsu ( Zapis konfiguracji interfejsu szeregowego ).

### 6.2 INTERFEJS WIEGAND

Czytnik po skonfigurowaniu do pracy w trybie WIEGAND wysyła unikalny numer ID przeczytanej karty zgodnie z protokołem Wiegand o parametrach:

Czas trwania impulsu (poziom L) 100us

Odstęp pomiędzy impulsami (poziom H) 1ms



Czytnik MW-R7x pozwala na zmianę długości ramki WIEGAND oraz na wybór części ID karty, która ma zostać wysłana po magistrali.

Przykłady:

ID karty = 0x123456789A = 0b0001001000110100010101100111100010011010

Parametry WIEGAND	ID karty / odpowiadająca ramka WIEGAND	
P1=26, P2=0	0b0001001000110100010101100111100010011010 P000100100011010001010110N	ID karty Ramka WIEGAND
P1=37, P2=0	0b0001001000110100010101100111100010011010 P00010010001101000101011001111000100N	ID karty Ramka WIEGAND
P1=26, P2=1	0b0001001000110100010101100111100010011010 P010101100111100010011010N	ID karty Ramka WIEGAND

P, N – bity parzystości

Inny format, np. WIEGAND można uzyskać poprzez zmianę konfiguracji ( Zapis konfiguracji interfejsu szeregowego ).

### 6.3 INTERFEJS RS232(TTL) / RS485 / CAN

Czytnik MW-R7x nasłuchuje komend wysyłanych poprzez interfejs:

- RS232 (TTL)
- RS485/CAN (w zależności od konfiguracji)

### 6.3.1 PROTOKÓŁ KOMUNIKACJI DLA INTERFEJSU RS232(TTL) / 485

Przy komunikacji poprzez interfejs RS232 / RS485 wykorzystywany jest protokół NETRONIX lub protokół MODBUS RTU.

W niniejszej dokumentacji opis protokołu NETRONIX 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
----------	-------------------	---------------------------	-------------	-----

#### UWAGI:

- Pracę z protokołem NETRONIX przetestować można za pomocą narzędziowego, darmowego oprogramowania „FRAMER”.
- Do konfiguracji urządzenia można użyć darmowego oprogramowania NEFIG.

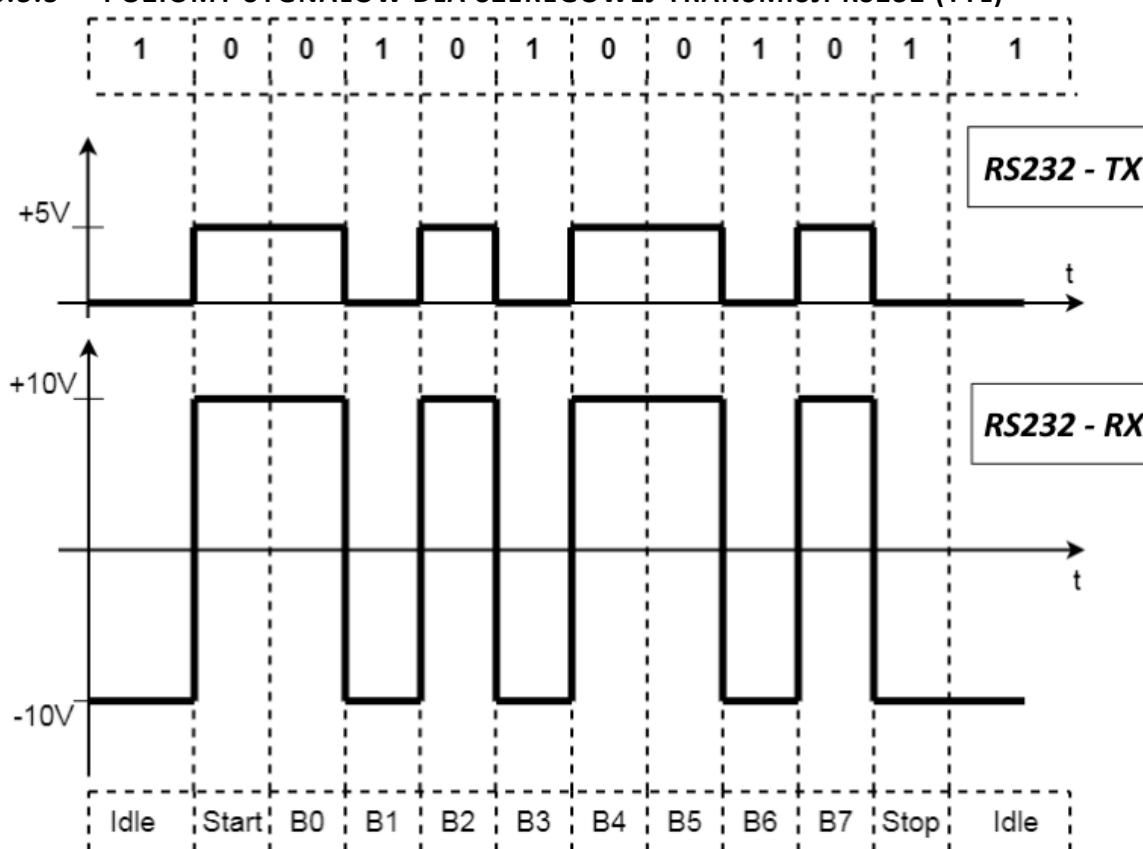
### 6.3.2 PROTOKÓŁ KOMUNIKACJI DLA INTERFEJSU CAN

Przy komunikacji poprzez interfejs szeregowy CAN wykorzystywana jest warstwa pośrednicząca, umożliwiająca przesyłanie ramek zgodnych z protokołem NETRONIX przy użyciu ramek danych w formacji CAN 2.0B.

Pełna specyfikacja protokołu wykorzystywanego przy komunikacji poprzez interfejs CAN znajduje się w dokumentacji „Protokół CAN NX”.

*Do komunikacji z wykorzystaniem interfejsu CAN, producent zaleca użycie konwerterów COTER-ED. Konwertery te mają zaimplementowaną warstwę pośredniczącą, co sprawia, że z urządzeniami MW-R7x można się komunikować w sposób identyczny, jak przy użyciu interfejsu RS485.*

### 6.3.3 POZIOMY SYGNAŁÓW DLA SZEREGOWEJ TRANSMISJI RS232 (TTL)



#### UWAGI:

*Do linii RX można podłączyć zarówno sygnał o standardowych poziomach (-10V..+10V) jak i o poziomach TTL (0..+5V).*

## 7 KOMENDY DOSTĘPNE DLA INTERFEJSU RS232(TTL) / RS485 / CAN – PROTOKÓŁ NETRONIX

### 7.1 KONFIGURACJA INTERFEJSU SZEREGOWEGO

#### 7.1.1 ZAPIS KONFIGURACJI INTERFEJSU SZEREGOWEGO

Ramka rozkazu:

C_SetInterfaceConfig	P0, P1, P2, [P3]
----------------------	------------------

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_SetInterfaceConfig	Komenda zmiany ustawień interfejsu szeregowego	0x54
P0	Typ interfejsu	0 – RS232 1 – RS485 2 – 1-WIRE 3 – WIEGAND 4 – CAN
P1, P2, [P3]	<p>Parametry zależne od wartości pola P0:</p> <p><b>Dla Typ=0</b>            P1 – Adres logiczny (RS232) P1: 0x01 - 0xFE            P2 – Prędkość transmisji (RS232) P2: – Patrz Tabela 7.1            [P3] – Parametr opcjonalny. Wymuszenie inicjalizacji linii TX przy starcie urządzenia. Brak parametru nie zmienia obecnej wartości. P3: 0 – Nie inicjalizuj TX, 1 – inicjalizuj TX.</p> <p><b>Dla Typ=1 lub Typ=4</b>            P1 – Adres logiczny (RS485 / CAN) P1: 0x01 - 0xFE            P2 – Prędkość transmisji (RS485) P2: – Patrz Tabela 7.1            [P3] – Parametr opcjonalny. Przełącznik interfejsu RS485/CAN P3: 0 – RS485 (wartość domyślna), 1 - CAN</p> <p><b>Dla Typ=2</b>            P1 – ConfAdr (7 bajt ramki Dallas) P1: 0x00-0xFF            P2 – ConfFC (1 bajt ramki Dallas) P2: 0x00-0xFF</p> <p><b>Dla Typ=3</b>            P1 – Ilość bitów P1: 26 - 48            P2 – L/M. Przełącznik ten określa, która część ID karty zostanie wysłana w ramce WIEGAND P2: 0-1</p>	

Tabela 7.1 Prędkość interfejsu RS232

ID	Prędkość
0	1200 bps
1	2400 bps
2	4800 bps
3	9600 bps
4	19200 bps
5	38400 bps
6	57600 bps
7	115200 bps

Ramka odpowiedzi:

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

#### 7.1.2 ODCZYT KONFIGURACJI INTERFEJSU SZEREGOWEGO

Ramka rozkazu:

C_GetInterfaceConfig	P0
----------------------	----

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_GetInterfaceConfig	Komenda odczytu ustawień interfejsu szeregowego	0x56
P0	Typ interfejsu, którego konfigurację chcemy odczytać	0 – RS232 1 – RS485 2 – 1-WIRE 3 – WIEGAND 4 – CAN

Ramka odpowiedzi:

C_GetInterfaceConfig+1	P0, P1, P2
------------------------	------------

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_GetInterfaceConfig+1	Komenda odczytu ustawień interfejsu szeregowego	0x57
P0	Typ interfejsu	0 – RS232 1 – RS485 2 – 1-WIRE 3 – WIEGAND 4 – CAN
P1, P2, [P3]	<p>Parametry zależne od wartości pola P0:</p> <p><b>Dla Typ=0</b> P1 – Adres logiczny (RS232) P1: 0x01 - 0xFE P2 – Prędkość transmisji (RS232) P2: – Patrz Tabela 7.1</p> <p><b>Dla Typ=1 lub Typ=4</b> P1 – Adres logiczny (RS485 / CAN) P1: 0x01 - 0xFE P2 – Prędkość transmisji (RS485) P2: – Patrz Tabela 7.1 [P3] – Parametr opcjonalny. Przełącznik interfejsu RS485/CAN P3: 0 – RS485 (wartość domyślna), 1 - CAN</p> <p><b>Dla Typ=2</b> P1 – ConfAdr (7 bajt ramki Dallas) P1: 0x00-0xFF P2 – ConfFC (1 bajt ramki Dallas) P2: 0x00-0xFF</p> <p><b>Dla Typ=3</b> P1 – Ilość bitów P1: 26 - 37 P2 – L/M. Przełącznik ten określa, która część ID karty zostanie wysłana w ramce WIEGAND P2: 0-1</p>	

## 7.2 ROZKAZY KOMUNIKACJI Z TRANSPONDERAMI

### 7.2.1 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.

#### 7.2.1.1 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
----------	-------------------	-------------	-----

#### 7.2.1.2 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. W czytniku można zapisać do 32 różnych kluczy.	0x00...0x1f

Ramka odpowiedzi:

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

#### 7.2.1.3 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. W czytniku można zapisać do 32 różnych kluczy.	0x00...0x1f
Key0..Key15	16-bajtowy klucz	

Ramka odpowiedzi:

nagłówek	C_DesSaveKey +1	KodOperacji	CRC
----------	-----------------	-------------	-----

### 7.2.2 ROZKAZY WSPÓLNE DO KOMUNIKACJI Z TRANSPONDERAMI

#### 7.2.2.1 ZAŁĄCZANIE I WYŁĄCZANIE POLA CZYTNIKA

Ramka rozkazu:

C_TurnOnAntennaPower	State
----------------------	-------

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_TurnOnAntennaPower	Załączanie i wyłączanie pola czytnika	0x10
State	Stan	0x00 – wyłączenie pola 0x01 – załączanie pola

Ramka odpowiedzi:

C_TurnOnAntennaPower +1		KodOperacji
-------------------------	--	-------------

### 7.2.2.2 WYSELEKJONOWANIE JEDNEGO TRANSPONDERA Z WIELU

Ramka rozkazu:

C_Select	
----------	--

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_Select	Odczyt ID	0x12

Ramka odpowiedzi:

C_Select +1	Coll, TType, ID1.....IDn	KodOperacji
-------------	--------------------------	-------------

Gdzie:

Nazwa parametru	Opis parametru	znaczenie
Coll	Informacja o kolizji (tylko transpondery HITAG)	0 – brak kolizji 1 – kolizja dwóch lub więcej transponderów
TType	Informacja o typie transpondera, z którego pochodzi odczytany numer ID	0x50 – S50 0x70 – S70 0x10 – Ultra Light 0xDF – Des Fire
ID1...IDn	Unikalny numer transpondera	ID1 – LSB, IDn – MSB

### 7.2.2.3 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
----------	----------	--	-------------	-----

## 7.2.3 ROZKAZY DO KOMUNIKACJI Z TRANSPONDERAMI MIFARE CLASSIC

### 7.2.3.1 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 0xBB – klucz typu B

DKNo	Numer dynamicznego klucza	0x00
------	---------------------------	------

Ramka odpowiedzi:

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

### 7.2.3.2 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 0xBB – klucz typu B
SKNo	Numer statycznego klucza	0x00...0x1F

Ramka odpowiedzi:

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

## 7.2.3.3 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	

## 7.2.3.4 ZAPIS ZAWARTOŚCI BLOKU TRANSPONDERA

Ramka rozkazu:

nagłówek	C_WriteBlock	BlockNo, Data1..... Data16	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
----------	-----------------	--	-------------	-----

## 7.2.3.5 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
----------	----------------	--	-------------	-----

## 7.2.3.6 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	**NumeracjaBlokówISektorów

	dla działania systemu a użytkownik sam może/powinien zobaczyć kopię Wartości.	
Value1...4	Wartość zapisywana do bloku transpondera	dowolne

Ramka odpowiedzi:

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

### 7.2.3.7 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

### 7.2.3.8 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
----------	---------------------	--	-------------	-----

### 7.2.3.9 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
----------	--------------------	--	-------------	-----

## 7.2.4 ROZKAZY DO KOMUNIKACJI Z TRANSPONDERAMI MIFARE ULTRALIGHT

### 7.2.4.1 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
----------	------------------	--	-------------	-----

### 7.2.4.2 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

### 7.2.4.3 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
----------	---------------	--	-------------	-----

## 7.2.5 ROZKAZY DO KOMUNIKACJI Z TRANSPONDERAMI MIFARE PLUS

### 7.2.5.1 ROZKAZY POZIOMU SŁO

#### 7.2.5.1.1 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	Dwubajttwowy 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
----------	----------------	--	-------------	-----

#### 7.2.5.1.2 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
----------	----------------	--	-------------	-----

#### 7.2.5.2 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

##### 7.2.5.2.1 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
----------	----------------	--	-------------	-----

#### 7.2.5.3 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	Dwubajttwowy numer bloku lub klucza do zapisania	0x9002 – przejście do poziomu SL2 0x9003 – przejście do poziomu SL3

		0x8000 – sprawdzenie oryginalności transpondera
KeyIdx	Indeks klucza AES zapisanego w czytniku	0x00-0x1F

Ramka odpowiedzi:

nagłówek	C_MfPlusCMD +1		KodOperacji	CRC
----------	----------------	--	-------------	-----

#### 7.2.5.4 ROZKAZY POZIOMU SL3

##### 7.2.5.4.1 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		0x3E
CID	Identyfikator CID	0x00

Ramka odpowiedzi:

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

##### 7.2.5.4.2 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 0x00-0x27 – karta Plus 4k
KeyType	Typ klucza	0xAA – klucz A 0xBB – klucz B
KeyIdx	Indeks klucza AES zapisanego w czytniku	0x00-0x1F

Ramka odpowiedzi:

nagłówek	C_MfPlusCMD +1		KodOperacji	CRC
----------	----------------	--	-------------	-----

##### 7.2.5.4.3 ODCZYT ZAWARTOŚCI BLOKU TRANSPONDERA

Ramka rozkazu:

nagłówek	C_MfPlusCMD	read_cmd, block		CRC
----------	-------------	-----------------	--	-----

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_MfPlusCMD	Komenda obsługi MFPlus	0x3A
read_cmd	Typ procedury odczytu:	0x30-0x33



	cmd.	MAC on command	MAC on response	Plain /encrypted	
	0x30	Yes	No	Encrypted*	
	0x31	Yes	Yes	Encrypted*	
	0x32	Yes	No	Plain	
	0x33	Yes	Yes	Plain	
block	Numer bloku do odczytu				0-3 dla sektorów<32 0-15 dla sektorów>32

\*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	

#### 7.2.5.4.4 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
	cmd.	MAC on command
		MAC on response
		Plain /encrypted
	0xA0	Yes No Encrypted*
	0xA1	Yes Yes Encrypted*
	0xA2	Yes No Plain
	0xA3	Yes Yes Plain
block	Numer bloku do odczytu	0-3 dla sektorów<32 0-15 dla sektorów>32
data0..data15	Dane do zapisu bloku transpondera	

\*tylko transpondery Plus X

Ramka odpowiedzi:

nagłówek	C_MfPlusCMD +1		KodOperacji	CRC
----------	----------------	--	-------------	-----

#### 7.2.5.5 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

### 7.2.6 OBSŁUGA TRANSPONDERÓW DESFIRE, DESFIRE EV1

#### 7.2.6.1 AUTORYZACJA, LOGOWANIE DO AKTUALNIE WYSELEKCYJONOWANEJ 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 : 0x0A – DES 0xAA – AES 0x3A – 3DES	0x0A, 0xAA, 0x3A

Ramka odpowiedzi:

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

### 7.2.6.2 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 bajta 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, 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

### 7.2.6.3 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	<ul style="list-style-type: none"> <li>Jeśli zmieniany klucz nie jest tym, którym nastąpiła aktualna autoryzacja, podajemy indeks aktualnego klucza, który będzie zmieniany</li> <li>Jeśli zmieniany klucz jest tym samym, którym nastąpiła aktualna autoryzacja, parametr ten pozostawiamy pusty</li> </ul>	0x00..0x13

Ramka odpowiedzi:

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

### 7.2.6.4 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	<i>Bit3..bit0:</i> Liczba kluczy przypisanych do danej aplikacji <i>Bit7..Bit6:</i> 00 – autoryzacja DES dla całej aplikacji 10- autoryzacja AES dla całej aplikacji	0x00..0x0D

Ramka odpowiedzi:

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

Struktura bajta 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

### 7.2.6.5 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
----------	-------------------	--	-------------	-----

### 7.2.6.6 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

### 7.2.6.7 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
----------	------------------	-------------	-----

### 7.2.6.8 FORMATOWANIE TRANSPONDERA

Ramka rozkazu:

nagłówek	C_DesFormatPICC (0x70)	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
----------	--------------------	-------------	-----

### 7.2.6.9 INICJALIZACJA PROTOKOŁU TRANSMISJI Z TRANSPONDERAMI DESFIRE

Ramka rozkazu:

nagłówek	C_DesInitProtocol (0x3E)	CID	CRC
----------	--------------------------	-----	-----

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_DesInitProtocol	Komenda formatowania transpondera	0x3E
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
----------	----------------------	-------------	-----

### 7.2.6.10 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

### 7.2.6.11 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_DesGetApplDs +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

### 7.2.6.12 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: 0x01 – nieszyfrowana 0x03 – szyfrowana DES	0x00,0x03
AccRight1..2	Prawa dostępu do pliku, patrz tabela poniżej	0x00..0xff
FileSize1..3	3 bajtowa wielkość pliku w bajtach, w kolejności LSB..MSB	0x00-0xff

Bajty określające prawa dostępu:

15	12	11	8	7	4	3	0
<b>Read Access</b>		<b>Write Access</b>		<b>Read &amp; Write Access</b>		<b>Change Right Access</b>	
MBS				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
----------	--------------------------	--	-------------	-----

7.2.6.13 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: 0x01 – nieszyfrowana 0x03 – szyfrowana DES	0x00,0x03
AccRight1..2	Prawa dostępu do pliku	0x00..0xff
FileSize1..3	3 bajtowa wielkość pliku w bajtach 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.7.2.6.14 TWORZENIE PLIKÓW TYPU *LINEAR/CYCLIC RECORD FILES*

Ramka rozkazu:

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

Gdzie:

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

Ramka odpowiedzi:

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

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

## 7.2.6.15 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
----------	-------------------	-------------	-----

## 7.2.6.16 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: 0x01 – nieszyfrowana 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*7.2.6.17 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
----------	------------------	---------	-------------	-----

7.2.6.18 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
----------	------------------	--	-------------	-----

7.2.6.19 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	0x00-0xFF



	mniejsza niż wielkość pojedynczego rekordu)	
Data1..58	Dane, które zamierzamy zapisać do pliku, (jednorazowo zapisać można do 58 bajtów, 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.

### 7.2.6.20 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..0xFF
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

### 7.2.6.21 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..0xFF

Ramka odpowiedzi:

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

**UWAGA:**

Operacja ta musi zakończyć się wydaniem komendy C\_DesCommit.

### 7.2.6.22 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
----------	---------------	--	-------------	-----



### 7.2.6.23 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
----------	-----------------	--	-------------	-----

### 7.2.7 TRANSMISJA DANYCH I-BLOCK PROTOKOŁU T=CL ISO14443-4

Komenda ta umożliwi 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
----------	------------------	------	--	-------------	-----

### 7.2.8 OBSŁUGA TRANSPONDERÓW RODZINY I-CODE SLI

#### 7.2.8.1 ODCZYT NUMERU ID TRANSPONDERA I-CODE SLI

Ramka rozkazu:

nagłówek	C_Inventory			CRC
----------	-------------	--	--	-----

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_Inventory	Odczyt numeru ID	0x04

Ramka odpowiedzi:

Nagłówek	C_Inventory +1	0,CardType,ID1...ID8		KodOperacji	CRC
----------	----------------	----------------------	--	-------------	-----

#### 7.2.8.2 ODCZYT STRONY TRANSPONDERA SLI

Ramka rozkazu:

Nagłówek	C_SLIReadPage	PageAdr		CRC
----------	---------------	---------	--	-----

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_SLIReadPage	Odczyt zawartości stron w Mifare UL	0x2C
PageAdr	Adres strony zgodny z obsługiwany typem transpondera	

Ramka odpowiedzi:

Nagłówek	C_SLIReadPage +1	Data1...4		KodOperacji	CRC
----------	------------------	-----------	--	-------------	-----

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
Data1...4	Odczytane dane.	dowolne

#### 7.2.8.3 ZAPIS ZAWARTOŚCI STRONY W SLI

Ramka rozkazu:

nagłówek	C_SLIWritePage	PageAdr, Data1...4	CRC
----------	----------------	--------------------	-----

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_SLIWritePage	Zapis zawartości strony w SLI	0x2E
PageAdr	Numer strony w transponderze	
Data1...4	Dane jakie mają być zapisane	dowolne

Ramka odpowiedzi:

nagłówek	C_SLIWritePage +1	KodOperacji	CRC
----------	-------------------	-------------	-----

## 7.2.9 MIFARE APPLICATION DIRECTORY - MAD

### 7.2.9.1 FORMATOWANIE KARTY MAD

Ramka rozkazu:

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

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_FormatMad 0xa8	Formatowanie do MAD	0xa8
Type	1 - MAD1 (15sektorów) 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

### 7.2.9.2 DODANIE APLIKACJI DO KATALOGU MAD

Ramka rozkazu:

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

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_AddApplication 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 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

### 7.2.9.3 WYSZUKANIE SEKTORA DLA DANEJ APLIKACJI

Ramka rozkazu:

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

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_GetSectorMad 0xac	Wyszukanie sektora	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)
- włączyć zasilanie anteny (komendą C\_TurnOnAntennaPower)
- wyselekcjonować kartę (komendą C\_Select)
- załogować 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

### 7.2.9.4 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 wyszukiwania 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

## 7.3 ŹRÓDŁA ORAZ WEJŚCIA I WYJŚCIA ELEKTRYCZNE

### 7.3.1 ZAPIS STANU ŹRÓDŁA RSX

Ramka rozkazu:

nagłówek	C_WriteSourceRSx	Source, State, [Time]	CRC
----------	------------------	-----------------------	-----

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_WriteSourceRSx	Zapis stanu źródła RSx	0x74
Source	Numer źródła RSx.	0x04-0x07
State	Żądany stan wyjścia	0x00 lub 0x01
[Time]	Parametr opcjonalny. Czas po jakim źródło RSx wróci do stanu 0 (x10ms)	0x00-0xFF

Ramka odpowiedzi:

nagłówek	C_WriteSourceRSx +1		KodOperacji	CRC
----------	---------------------	--	-------------	-----

### 7.3.2 ODCZYT STANU ŹRÓDŁA

Ramka rozkazu:

nagłówek	C_ReadSource	Source	CRC
----------	--------------	--------	-----

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_ReadSource	Odczyt stanu źródła	0x72
Source	Source	Patrz numer ID z Tabela 4.2 Źródła sygnałów

Ramka odpowiedzi:

nagłówek	C_ReadSource +1	State	KodOperacji	CRC
----------	-----------------	-------	-------------	-----

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_ReadSource+1	Odczyt stanu źródła	0x73
State	Wartość źródła	0x04-0x07

### 7.3.3 ZAPIS KONFIGURACJI PORTU

Ramka rozkazu:

C_SetIOConfig	IONo, Dir, P0
---------------	---------------

Gdzie:

**Jeżeli Konfigurujemy port jako wyjście:**

Nazwa parametru	Opis parametru	Zakres wartości
C_SetIOConfig	Zapis konfiguracji dowolnego portu	0x50
IONo	Numer portu IO, który ma być skonfigurowany	0x00..0x05
Dir	Kierunek portu	0x00 – wyjście
P0	Źródło sygnału sterującego	Patrz numer ID z Tabela 4.2 Źródła sygnałów

**Jeżeli Konfigurujemy port jako wejście:**

Nazwa parametru	Opis parametru	Zakres wartości
C_SetIOConfig	Zapis konfiguracji dowolnego portu	0x50
IONo	Numer portu IO, który ma być skonfigurowany	0x06 – 0x07
Dir	Kierunek portu.	1 – wejście
P0	Bajt określający sposób wyzwiania. Patrz rozdział: 4.4.5 Źródło PinINx	0 – niezanegowane 1 – zanegowane 2 – reakcja na zbocze rosnące 3 – reakcja na zbocze opadające

Nie wszystkie porty MW-R7x mają dowolny kierunek. W celu poprawnej konfiguracji należy dla danego portu podać poprawny kierunek.

Tabela 7.2 Spis istniejących portów, którymi można sterować w MW-R7x

Numer portu	kierunek	Opis
0	wyjście	Wyjście fizyczne PinOUT
1	wyjście	KOLOR0
2	wyjście	KOLOR1
3	wyjście	KOLOR2
4	wyjście	KOLOR3
5	wyjście	BUZZER
6	wejście	Wejście fizyczne PinIN0
7	wejście	Wejście fizyczne PinIN1

Ramka odpowiedzi:

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

### 7.3.4 ODCZYT KONFIGURACJI 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, którego konfiguracja ma być odczytana	0x00...0x07

Ramka odpowiedzi:

nagłówek	C_GetIOConfig +1	Dir, P0	KodOperacji	CRC
----------	------------------	---------	-------------	-----

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
Dir, P0	Parametry mają taką samą samo znaczenie jak przy zapisie komendzie C_SetIOConfig	

### 7.3.5 KONFIGURACJA BLOKU SIG\_A

Ramka rozkazu:

nagłówek	C_ConfigSIG_A	SigNo, [Function, In0, In1, In2]	CRC
----------	---------------	----------------------------------	-----

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_ConfigSIG_A	Zapis/odczyt konfiguracji bloku SIG_A	0x5C
SigNo	Numer bloku SIG_A, którego konfiguracja ma być odczytana / zapisana	0x00...0x03
Function	Parametr opcjonalny - jeżeli występuje, komenda powoduje zapis nowej konfiguracji. Określa rodzaj funkcji wykonywany przez blok SIG_A.	0 – funkcja OR 1 – funkcja AND
In1, In2, In3	Parametry opcjonalne - jeżeli występuje, komenda powoduje zapis nowej konfiguracji. Źródła sygnałów wejściowych	Patrz numer ID z Tabela 4.2 Źródła sygnałów

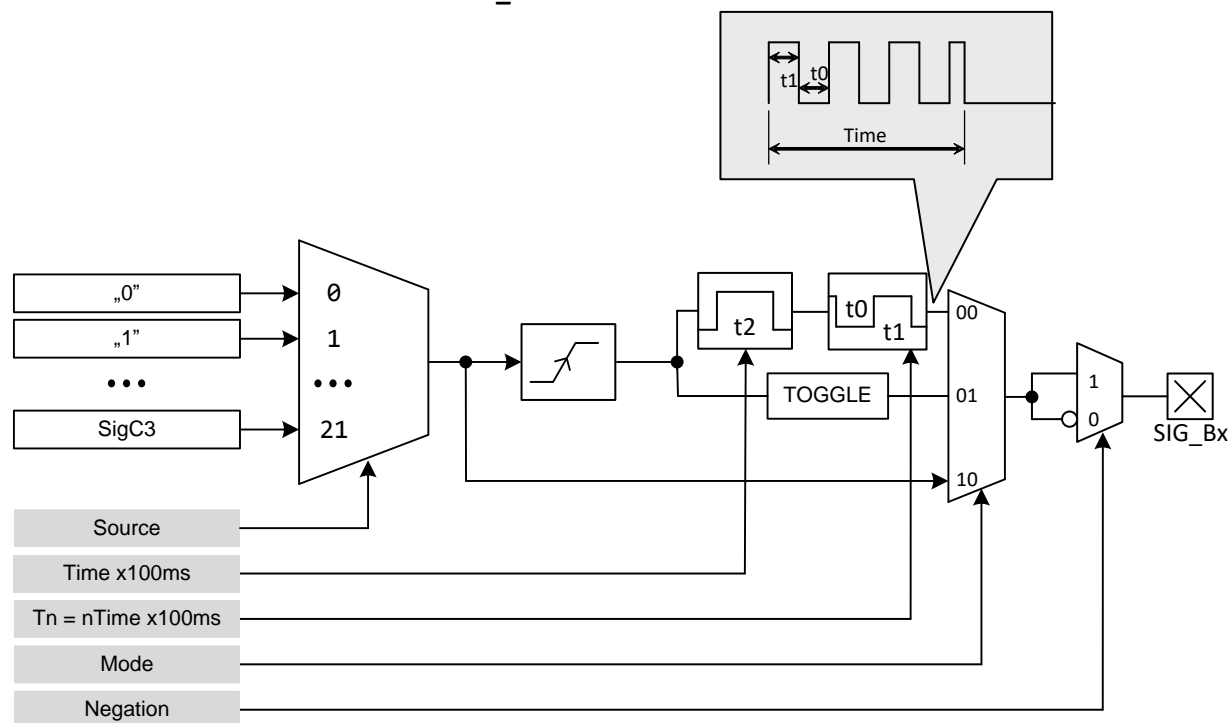
Ramka odpowiedzi:

nagłówek	C_ConfigSIG_A +1	Function, In0, In1, In2	KodOperacji	CRC
----------	------------------	-------------------------	-------------	-----

Gdzie:

Znaczenie parametrów odpowiedzi jest identyczne jak opisane wcześniej.

### 7.3.6 KONFIGURACJA BLOKU SIG\_B



Ramka rozkazu:

C_ConfigSIG_B	No, [Source, Mode, Negation, Time, 0Time, 1Time]
---------------	--

Parametry: *Source, Mode, Negation, Time, 0Time, 1Time* są opcjonalne i jeśli występują, to nastąpi zapisanie nowej konfiguracji.

Nazwa parametru	Opis parametru	Zakres wartości
C_ConfigSIG_B	Zapis/odczyt konfiguracji bloku SIG_B	0x60
No	Numer bloku SIG_B	0x00..0x03
Source	Źródło sygnału sterującego	Patrz numer ID z Tabela 4.2 Źródła sygnałów
Mode	Określa sposób zachowania wyjścia.	00 – generator fali prostokątnej 01 – zmiana stanu wyjścia na przeciwny względem poprzedniego stanu 10 – bezpośrednio
Negation	Negacja wyjścia.	0 – Wyjście zanegowane 1 – Wyjście wprost
Time	Czas podtrzymania stanu załączenia po ustaniu pobudzenia. Czas ten wyrażony jest jako: Podtrzymanie x 100ms  Podczas trwania czasu „Podtrzymanie” można skonfigurować wyjście potrafiące generować falę prostokątną. Czas jedynek i czas zera ustawiany jest następnymi parametrami 0Time oraz 1Time	0-255
0Time	czas logicznego zera	0-255
1Time	czas logicznej jedynek	0-255

Ramka odpowiedzi:

C_ConfigSIG_B+1	No, Source, Mode, Negation, Time, 0Time, 1Time
-----------------	--

Gdzie:

Znaczenie parametrów odpowiedzi jest identyczne jak opisane wcześniej.

### 7.3.7 KONFIGURACJA BLOKU SIG\_C

Ramka rozkazu:

C_ConfigSIG_C	No, [Source, Time]
---------------	--------------------

Parametry: *Source, Time* są opcjonalne i jeśli występują, to nastąpi zapisanie nowej konfiguracji.

Nazwa parametru	Opis parametru	Zakres wartości
C_ConfigSIG_C	Zapis/odczyt konfiguracji bloku SIG_B	0x62
No	Numer bloku SIG_C	0x00..0x03
Source	Źródło sygnału sterującego	Patrz numer ID z Tabela 4.2 Źródła sygnałów
Time	Czas filtrowania (x100ms)	0-255

Ramka odpowiedzi:

C_ConfigSIG_C+1	No, Source, Time
-----------------	------------------

Gdzie:

Znaczenie parametrów odpowiedzi jest identyczne jak opisane wcześniej.

### 7.3.8 KONFIGURACJA KOLORÓW

Ramka rozkazu:

nagłówek	C_ConfigLed	[C0, C1, C2, C3]	CRC
----------	-------------	------------------	-----

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_ConfigLed	Zapis/odczyt konfiguracji wyświetlanych kolorów	0x5E
[C0, C1, C2, C3]	Parametry opcjonalne - jeżeli występują, komenda powoduje zapis nowej konfiguracji. C0 – kod koloru0 o priorytecie 1 (najwyższym) C1 – kod koloru1 o priorytecie 2 C2 – kod koloru2 o priorytecie 3 C3 – kod koloru3 o priorytecie 4 (najniższym)	Patrz: <i>Tabela 4.1 Tabela kodów kolorów</i>

Ramka odpowiedzi:

nagłówek	C_ConfigLed +1	C0, C1, C2, C3	KodOperacji	CRC
----------	----------------	----------------	-------------	-----

Gdzie:

Znaczenie parametrów odpowiedzi jest identyczne jak opisane wcześniej.

## 7.4 HASŁO DOSTĘPU

### 7.4.1 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
----------	----------------	-------------	-----

### 7.4.2 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
----------	---------------------	--	-------------	-----

### 7.4.3 WYLOGOWANIE Z CZYTNIKA

Rozkaz ten dezaktualizuje podane ostatnio hasło.

Ramka rozkazu:

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

Gdzie:

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

Ramka odpowiedzi:

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

## 7.5 MECHANIZM AUTOREADER

### 7.5.1 ZAPIS KONFIGURACJI AUTOMATU

Rozkaz C\_SetAutoReaderConfig konfiguruje sposób pracy automatu odczytującego unikalny numer transpondera.

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

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,, [AModeParam], Abuzz, AMulti, Alnterface	CRC
----------	-----------------------	---	-----

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_SetAuto-ReaderConfig	Zapis konfiguracji automatu	0x58
ATrig	Definiuje kiedy automat odczytów UID ma pracować	0 - automat wyłączony na stałe 1 - automat załączony na stałe 2 - załącza się automatycznie, gdy brak transmisji na RS przez czas dłuższy niż AOfflineTime 3 - załącza się automatycznie, gdy brak wywołań rozkazów komunikacji z transponderem



		przez czas dłuższy niż AOfflineTime				
AOfflineTime	Czas braku transmisji na RS/USB $T = AOfflineTime * [100 \text{ ms}]$ Brak transmisji może dotyczyć dowolnych rozkazów (Atrig=2), lub rozkazów komunikacji z transponderem (Atrig=3).  Rozkazy komunikacji z transponderem to: C_TurnOnAntennaPower C_Select	0x00...0xff				
ASerial	Automatyczne wysyłanie numeru transpondera UID po automatycznym odczytaniu go z transpondera	0 - nigdy 1 - tylko za pierwszym przyłożeniem transpondera 2 - wysyła wszystkie				
AMode	Bajt konfiguracyjny określający format wysyłanego ID.  Format: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th style="text-align: left;">MSB</th> <th style="text-align: right;">LSB</th> </tr> </thead> <tbody> <tr> <td>I E F&lt;1,0&gt; C&lt;1,0&gt; D ID</td> <td></td> </tr> </tbody> </table> UWAGA: Bity E oraz F<1,0> mają znaczenie tylko dla AInterface=0 lub AInterface=1.  Bity C1, C0, D mają znaczenie tylko dla formatu ASCII (F<1,0>=1)	MSB	LSB	I E F<1,0> C<1,0> D ID		I=1 - Numer w odwrotnej kolejności  E=1 - informacja rozszerzona o sygnalizację kolizji oraz typ karty  F<1,0>=0 - ID w formacie ramki Nertonic F<1,0>=1 - ID w formacie ASCII F<1,0>=2 - ID w formacie binarnym F<1,0>=3 - ID w formacie ASCII (DEC)  C<1,0>=0 - Bez znaku końca linii C<1,0>=1 - Znak końca CR C<1,0>=2 - Znak końca LF C<1,0>=3 - Znak końca CRLF  D=1 - Konwersja na format dziesiętny, tylko dla trybu ASCII  ID – informacja rozszerzona o adres czytnika ustawiony dla magistrali RS485
MSB	LSB					
I E F<1,0> C<1,0> D ID						
[AModeParam]	Parametr opcjonalny. Jeśli nie występuje, to jego wartość jest ustalana na 0.	Dla AMode.F<1,0>=3 Określa ilość znaków wysyłanego ID.				
ABuzz	Automatyczne sygnalizowanie odczytu za pomocą buzzera po automatycznym odczytaniu UID'u z transpondera.	0 - nigdy 1 - tylko za pierwszym przyłożeniem transpondera 2 - sygnalizuje wszystkie				
AMulti	Tryb odczytu wielu typów transponderów Format: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th style="text-align: left;">MSB</th> <th style="text-align: right;">LSB</th> </tr> </thead> <tbody> <tr> <td>- - - - S I - M</td> <td></td> </tr> </tbody> </table>	MSB	LSB	- - - - S I - M		M – transpondery z rodziny MIFARE I – iClass (CSN) S – I-CODE (ISO15693)
MSB	LSB					
- - - - S I - M						
AInterface	Wybór interfejsu, po którym wysyłane jest odczytane ID przez automat autoreadera	0 – RS232 1 – RS485 / CAN <sup>(1)</sup> 2 – 1-WIRE 3 – WIEGAND 4 – RS485 / CAN <sup>(1)</sup>				

(1) – w zależności od tego, który interfejs jest ustawiony jako aktywny

Ramka odpowiedzi:

nagłówek	C_SetAutoReaderConfig +1		KodOperacji	CRC
----------	--------------------------	--	-------------	-----

## 7.5.2 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, AMode, Abuzz, AMulti	ASerial,	KodOperacji	CRC

Gdzie:

Znaczenie parametrów odpowiedzi jest identyczne jak opisane wcześniej.

## 7.6 OBSŁUGA ID ZAPISANYCH W PAMIĘCI TRANSPONDERA

Czytnik MW-R7 umożliwia odczyt ID zapisanego w pamięci transpondera przez użytkownika.

### 7.6.1 ZAPIS/ODCZYT KONFIGURACJI

Ramka rozkazu:

nagłówek	C_ConfigUserID	[CardType, ID_Len, ID_Offset, Param[...]]	CRC

Gdzie:

Nazwa parametru	Opis parametru	Zakres wartości
C_ConfigUserID	Kod rozkazu	0xBA
CardType	Typ karty	0x50 – Mifare S50 0x70 – Mifare S70 0xDF – DESFire
ID_Len	Długość zapisanego ID	0x01...0x20
ID_Offset	Przesunięcie ID względem 1 bajta w sektorze.	0x20-ID_Len
<b>Parametry dla CardType=0x50 lub CardType=0x70 (Mifare Classic)</b>		
Param[0] – SecNo	Numer sektora, w którym zapisane jest ID	
Param[1] – KeyType	Typ klucza	0xAA – klucz A 0xBB – klucz B
Param[2] – SKBKeyNo	Numer klucza w pamięci kluczy statycznych, który jest wykorzystywany do logowania do sektora.	0-0x1F
<b>Parametry dla CardType=0xDF (DESFire)</b>		
Param[0] – FileNo	Numer pliku	0x00 – 0xFF
Param[1] – AuthType	Typ klucza	0x0A – DES 0x3A – 3DES 0xAA – AES
Param[2] – KeyNo	Numer pod jakim klucz jest zapisany w pamięci	0-7
Param[3..5] – AppId	3 bajtowe ID aplikacji	0x000000 – 0xFFFFFFFF

Odpowiedź:

	C_ConfigUserID+1	CardType, ID_Len, ID_Offset, Param[]	KodOperacji	

Znaczenie pól takie samo jak powyżej.

## 7.7 ROZKAZY POZOSTAŁE

### 7.7.1 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

### 7.7.2 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.

## 7.8 ZNACZENIE KODÓW OPERACJI W RAMKACH ODPOWIEDZI

Tabela 7.3 Kody operacji

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_NoACKFromSlave	Brak wewnętrznej komunikacji	0x22
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
<b>Kody operacji związane z transponderami DESFIRE</b>		
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_DesPermissionDenied	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

## 8 PROTOKÓŁ MODBUS RTU

### UWAGA:

Przy korzystaniu z protokołu MODBUS należy zmienić konfigurację urządzenia tak, aby samo nie wysyłało odczytanego ID. W tym celu należy do rejestru o adresie 1022 (ASerial) zapisać wartość 0.

### 8.1 OBSŁUGIWANE FUNKCJE PROTOKOŁU MODBUS

Funkcja	Opis
0x01	Read Coils
0x02	Read Discrete Inputs
0x03	Read Holding Regs
0x04	Read Input Regs
0x05	Write Single Coil
0x06	Write Single Reg
0x10	Write Multiple Regs

### 8.2 ADRESY MODBUS

#### 8.2.1 ADRESY DO ODCZYTU ID KARTY

Adres	Typ	R/W	Opis
997	Holding Reg	R	B<15:8> - Typ transpondera B<7:0> - Ilość kolizji
998	Holding Reg	R	Długość ID
999	Holding Reg	R	Licznik czasu od ostatniego odczytu (x100ms)
1000	Holding Reg	R	Kod transpondera [0]
1001	Holding Reg	R	Kod transpondera [1]
1002	Holding Reg	R	Kod transpondera [2]
1003	Holding Reg	R	Kod transpondera [3]
1004	Holding Reg	R	Kod transpondera [4]
1005	Holding Reg	R	Kod transpondera [5]
1006	Holding Reg	R	Kod transpondera [6]
1007	Holding Reg	R	Kod transpondera [7]

#### 8.2.2 ADRESY DO ODCZYTU/ZAPISU KONFIGURACJI AUTOREADERA

Adres	Typ	R/W	Opis
1020	Holding Reg	R/W	ATrig
1021	Holding Reg	R/W	AOfflineTimer
1022	Holding Reg	R/W	ASerial
1023	Holding Reg	R/W	B<15:8> - AModeParam, B<7:0> - AMode
1024	Holding Reg	R/W	ABuzz
1025	Holding Reg	R/W	AMulti
1026	Holding Reg	R/W	AInterface

#### 8.2.3 ADRESY DO ODCZYTU/ZAPISU KONFIGURACJI INTERFEJSÓW RS232 / RS485

Adres	Typ	R/W	Opis
1030	Holding Reg	R/W	Adres na magistrali RS232
1031	Holding Reg	R/W	Prędkość transmiji na magistrali RS232
1032	Holding Reg	R/W	Adres na magistrali RS485
1033	Holding Reg	R/W	Prędkość trensmisji na magistrali RS485

#### 8.2.4 ADRESY DO ODCZYTU/ZAPISU KONFIGURACJI BLOKÓW SIGA

Adres	Typ	R/W	Opis
1040	Holding Reg	R/W	SigA_0 – Function
1041	Holding Reg	R/W	SigA_0 – In0
1042	Holding Reg	R/W	SigA_0 – In1

1043	Holding Reg	R/W	SigA_0 – In2
1044	Holding Reg	R/W	SigA_1 – Function
1045	Holding Reg	R/W	SigA_1 – In0
1046	Holding Reg	R/W	SigA_1 – In1
1047	Holding Reg	R/W	SigA_1 – In2
1048	Holding Reg	R/W	SigA_2 – Function
1049	Holding Reg	R/W	SigA_2 – In0
1050	Holding Reg	R/W	SigA_2 – In1
1051	Holding Reg	R/W	SigA_2 – In2
1052	Holding Reg	R/W	SigA_3 – Function
1053	Holding Reg	R/W	SigA_3 – In0
1054	Holding Reg	R/W	SigA_3 – In1
1055	Holding Reg	R/W	SigA_3 – In2

### 8.2.5 ADRESY DO ODCZYTU/ZAPISU KONFIGURACJI BLOKÓW SIGB

Adres	Typ	R/W	Opis
1060	Holding Reg	R/W	SigB_0 – Source
1061	Holding Reg	R/W	SigB_0 – Mode
1062	Holding Reg	R/W	SigB_0 – Negation
1063	Holding Reg	R/W	SigB_0 – Time
1064	Holding Reg	R/W	SigB_0 – 0Time
1065	Holding Reg	R/W	SigB_0 – 1Time
1066	Holding Reg	R/W	SigB_1 – Source
1067	Holding Reg	R/W	SigB_1 – Mode
1068	Holding Reg	R/W	SigB_1 – Negation
1069	Holding Reg	R/W	SigB_1 – Time
1070	Holding Reg	R/W	SigB_1 – 0Time
1071	Holding Reg	R/W	SigB_1 – 1Time
1072	Holding Reg	R/W	SigB_2 – Source
1073	Holding Reg	R/W	SigB_2 – Mode
1074	Holding Reg	R/W	SigB_2 – Negation
1075	Holding Reg	R/W	SigB_2 – Time
1076	Holding Reg	R/W	SigB_2 – 0Time
1077	Holding Reg	R/W	SigB_2 – 1Time
1078	Holding Reg	R/W	SigB_3 – Source
1079	Holding Reg	R/W	SigB_3 – Mode
1080	Holding Reg	R/W	SigB_3 – Negation
1081	Holding Reg	R/W	SigB_3 – Time
1082	Holding Reg	R/W	SigB_3 – 0Time
1083	Holding Reg	R/W	SigB_3 – 1Time

### 8.2.6 ADRESY DO ODCZYTU/ZAPISU KONFIGURACJI BLOKÓW SIGC

Adres	Typ	R/W	Opis
1090	Holding Reg	R/W	SigC_0 – Source
1091	Holding Reg	R/W	SigC_0 – Time
1092	Holding Reg	R/W	SigC_1 – Source
1093	Holding Reg	R/W	SigC_1 – Time
1094	Holding Reg	R/W	SigC_2 – Source
1095	Holding Reg	R/W	SigC_2 – Time
1096	Holding Reg	R/W	SigC_3 – Source
1097	Holding Reg	R/W	SigC_3 – Time

### 8.2.7 ADRESY DO ODCZYTU/ZAPISU KONFIGURACJI LED

Adres	Typ	R/W	Opis
1100	Holding Reg	R/W	C0 – kod koloru 0
1101	Holding Reg	R/W	C1 – kod koloru 1
1102	Holding Reg	R/W	C2 – kod koloru 2

<b>1103</b>	Holding Reg	R/W	C3 – kod koloru 3
-------------	-------------	-----	-------------------

### 8.2.8 ADRESY DO ODCZYTU/ZAPISU KONFIGURACJI I/O

Adres	Typ	R/W	Opis
<b>1104</b>	Holding Reg	R/W	Source – PinOut
<b>1105</b>	Holding Reg	R/W	Source – Colour0
<b>1106</b>	Holding Reg	R/W	Source – Colour1
<b>1107</b>	Holding Reg	R/W	Source – Colour2
<b>1108</b>	Holding Reg	R/W	Source – Colour3
<b>1109</b>	Holding Reg	R/W	Source – Buzzer
<b>1110</b>	Holding Reg	R/W	Source – PinIn0
<b>1111</b>	Holding Reg	R/W	Source – PinIn1

### 8.2.9 ADRESY DO ODCZYTU/ZAPISU KONFIGURACJI MECHANIZMU USERID

Adres	Typ	R/W	Opis
<b>1150</b>	Holding Reg	R/W	CardType
<b>1151</b>	Holding Reg	R/W	ID_Len
<b>1152</b>	Holding Reg	R/W	ID_Offset
<b>1153</b>	Holding Reg	R/W	Param[0]
<b>1154</b>	Holding Reg	R/W	Param[1]
<b>1155</b>	Holding Reg	R/W	Param[2]
<b>1156</b>	Holding Reg	R/W	Param[3]
<b>1157</b>	Holding Reg	R/W	Param[4]
<b>1158</b>	Holding Reg	R/W	Param[5]

Opis rejestrów jest taki sam jak dla komendy [C.ConfigUserID](#).

### 8.2.10 ADRESY DO ZAPISU WARTOŚCI ŹRÓDEŁ RSX

Adres	Typ	R/W	Opis
<b>1200</b>	Holding Reg	W	RS0
<b>1201</b>	Holding Reg	W	RS1
<b>1202</b>	Holding Reg	W	RS2
<b>1203</b>	Holding Reg	W	RS3

### 8.2.11 ADRESY DO ODCZYTU WARTOŚCI ŹRÓDEŁ

Adres	Typ	R/W	Opis
<b>100</b>	Coil / Discrete Inputs	R	„0”
<b>101</b>	Coil / Discrete Inputs	R	„1”
<b>102</b>	Coil / Discrete Inputs	R	Przycisk
<b>103</b>	Coil / Discrete Inputs	R	Każda karta
<b>104</b>	Coil / Discrete Inputs	R	RS_0
<b>105</b>	Coil / Discrete Inputs	R	RS_1
<b>106</b>	Coil / Discrete Inputs	R	RS_2
<b>107</b>	Coil / Discrete Inputs	R	RS_3
<b>108</b>	Coil / Discrete Inputs	R	PinIn0
<b>109</b>	Coil / Discrete Inputs	R	PinIn1
<b>110</b>	Coil / Discrete Inputs	R	SigA0
<b>111</b>	Coil / Discrete Inputs	R	SigA1
<b>112</b>	Coil / Discrete Inputs	R	SigA2
<b>113</b>	Coil / Discrete Inputs	R	SigA3
<b>114</b>	Coil / Discrete Inputs	R	SigB0
<b>115</b>	Coil / Discrete Inputs	R	SigB1
<b>116</b>	Coil / Discrete Inputs	R	SigB2
<b>117</b>	Coil / Discrete Inputs	R	SigB3
<b>118</b>	Coil / Discrete Inputs	R	SigC0

119	Coil / Discrete Inputs	R	SigC1
120	Coil / Discrete Inputs	R	SigC2
121	Coil / Discrete Inputs	R	SigC3

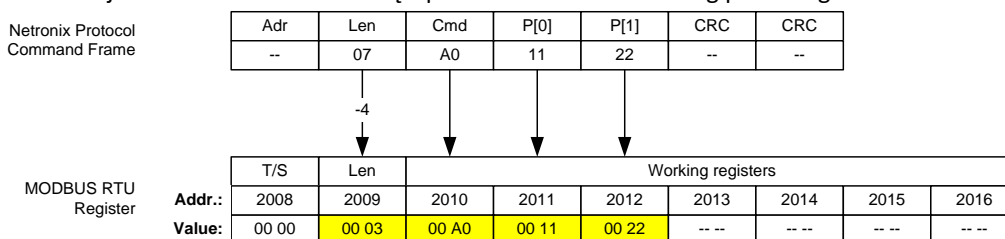
### 8.3 ENKAPSULACJA PROTOKOŁU NETRONIX W PROTOKOLE MODBUS RTU

Dowolną komendę z protokołu Netronix można wykonać wykorzystując odpowiednie rejestry z protokołu MODBUS.

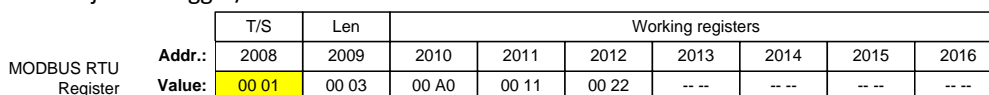
Adres	Typ	R/W	Nazwa	Opis
2008	Holding Reg	R/W	Trigger/Status	Rejestr ten służy do wyzwolenia przetwarzania komendy oraz do sprawdzania statusu przetwarzania. Dozwolone wartości: 0x0000 – Moduł w trybie Idle 0x0001 – Wyzwolenie przetwarzania 0x00EE – Błąd 0x00FF – Komenda wykonana. Odpowiedz znajduje się w rejestrach roboczych.
2009	Holding Reg	R/W	Len	Rejestr ten zawiera długość zapisanej komendy / długość odpowiedzi (ilość rejestrów zapisanych/do odczytania)
2010-2073	Holding Reg	R/W	Working registers	Rejestry te służą do zapisania komendy / odczytania odpowiedzi. Jeden rejestr przechowuje wartość 1 bajta komendy/odpowiedzi z ramki Netronix

#### 8.3.1 SCHEMAT POSTĘPOWANIA

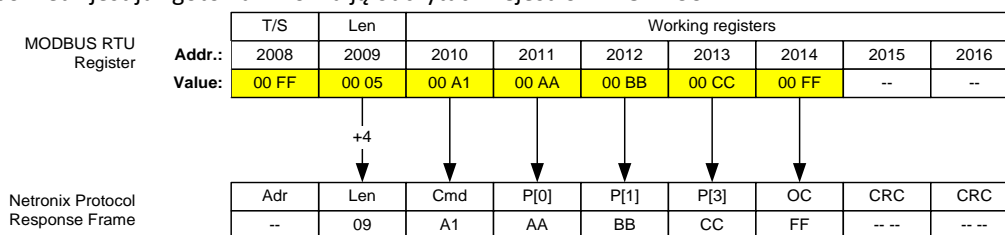
1. Zapisać do rejestrów MODBUS komendę z protokołu Netronix według poniższego schematu:



2. Zapisać do rejestru Trigger/Status wartość 0x0001.



3. Odczytywać rejestr Trigger/Status, aż pojawi się w nim wartość 0x00FF. Wartość 0x00FF oznacza, że odpowiedź jest już gotowa i można ją odczytać z rejestrów MODBUS.



#### 8.3.2 PRZYKŁAD UŻYCIA – ODCZYT WERSJI OPROGRAMOWANIA.

Założenia:

Adres logiczny czytnika (w protokole Netronix/Modbus RTU) – 0x01.

1. Należy określić jak powinna wyglądać ramka w protokole Netronix. Istotne jest pole Cmd oraz ewentualne parametry. Cała ramka, dla komendy odczytu wersji firmware wygląda następująco:

Adr	Len	Cmd	CRC	
0x01	0x05	0xFE	0xC6	0x14



Cmd = 0xFE  
Param – brak

1. Do rejestru **Len** należy wpisać ilość danych, a do rejestrów **Working registers** należy wpisać kod komendy (i ewentualne parametry):

```
RTU Tx > 01 10 07D8 0002 04 0001 00FE 0925
RTU Rx > 01 10 07D8 0002 C087
```

Len = 0001  
WorkingRegister[0] = 00FE

2. Do rejestru **Trigger/Status** należy wpisać wartość 0x0001. Spowoduje to wykonanie komendy zapisanej w rejestrach **Working Registers**.

```
RTU Tx > 01 06 07D7 0001 F946
RTU Rx > 01 06 07D7 0001 F946
```

Trigger/Status=0001

3. Następnie należy odczytywać rejestr **Trigger/Status**, aż do momentu odczytania wartości 0x00FF. Wartość 0x00FF oznacza że komenda została wykonana i w rejestrach **Working Registers** znajduje się odpowiedź.

```
RTU Tx > 01 03 07D7 0001 3546
RTU Rx > 01 03 02 00FF F804
```

Trigger/Status=00FF

4. Następnie należy odczytać wartość rejestru **Len**. Rejestr ten zawiera informację na temat ilości rejestrów, w których zapisana jest odpowiedź

```
RTU Tx > 01 03 07D8 0001 0545
RTU Rx > 01 03 02 0011 7848
```

Len=0011

5. W ostatnim kroku, należy odczytać Len pierwszych rejestrów roboczych.

```
RTU Tx > 01 03 07D9 0011 5549
RTU Rx > 01 03 22
00FF
004D 0057 002D 0052 0037 002D 0056 0033 002E 0032 002E 0041 0031 002E 0035
00FF
8E C6
```

00FF – Kod komendy+1 (odpowiedź na C\_FirmwareVersion)

004D 0057 002D ... 0031 002E 0035 – zwrócone wersja firmware – „MW-R7-v3.2.A.1.5”

00FF – Kod operacji (sukces)



## 9 POWRÓT DO USTAWIENÍ FABRYCZNYCH

Aby powrócić do ustawień fabrycznych należy w przedziale czasu od 3 do 10 sekund od uruchomienia urządzenia wcisnąć przycisk frontowy na czas ok. 5 sekund. Podczas powrotu do ustawień fabrycznych ustawiane są na stałe następujące parametry czytnika:

Tabela 9.1 Ustawienia fabryczne

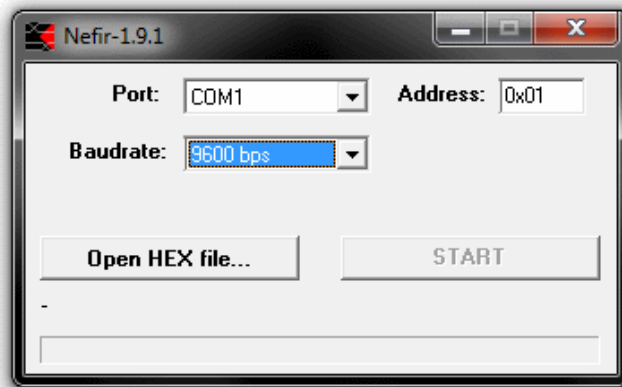
Nazwa parametru lub funkcjonalność	Wartość lub ustawienie	
<b>Interfejs</b>		
Interfejs RS232	Adres: 0x01 Prędkość: 0x03	9600bps
Interfejs RS485/CAN	Adres: 0x01 Prędkość: 0x03 Typ: 0x00	9600bps RS-485
Interfejs 1-WIRE	Rodzina: 0x01 Adres: 0x00	
Interfejs Wiegand	Ilość bitów 37	
<b>Odczyt transponderów</b>		
Autoreader	Triger: 0x02	
	Timeout: 0x14	2s
	Mode: 0xFF	wszystkie obsługiwane typy
	ASerial: 0x01	za pierwszym przyłożeniem
	AMode: 0x40	format Netronix, informacja rozszerzona o sygnalizację kolizji oraz typ karty
	ABuzzer: 0x01	za pierwszym przyłożeniem
	AMulti: 0x09	MIFARE + I-CODE
AInterface: 0x00	RS232	
<b>Wejścia/Wyjścia</b>		
Wejście PinIN0	Wyzwalanie: Stanem niskim	
Wejście PinIN1	Wyzwalanie: Stanem niskim	
Wyjście PinOUT	Sterowanie źródłem: Przycisk	
Wyjście Kolor0	Sterowanie źródłem: PinIN1	
Wyjście Kolor1	Sterowanie źródłem: Przycisk	
Wyjście Kolor2	Sterowanie źródłem: „0”	
Wyjście Kolor3	Sterowanie źródłem: „1”	
Wyjście Buzzer	Sterowanie źródłem: PinIN0	
<b>Ustawienie kolorów</b>		
Konfiguracja LED	C0: ZIELONY C1: NIEBIESKI C2: BIAŁY C3: CZERWONY	
<b>Bloki SIGNAL</b>		
SigA0	In0: „0”; In1: „0”; In2: „0”; Funkcja: OR	
SigA1	In0: „0”; In1: „0”; In2: „0”; Funkcja: OR	
SigA2	In0: „0”; In1: „0”; In2: „0”; Funkcja: OR	
SigA3	In0: „0”; In1: „0”; In2: „0”; Funkcja: OR	
SigB0	Source: „0”, Mode: 2, Negation: 1 Time: 0, Time0: 0, Time1: 0	
SigB1	Source: „0”, Mode: 2, Negation: 1 Time: 0, Time0: 0, Time1: 0	
SigB2	Source: „0”, Mode: 2, Negation: 1 Time: 0, Time0: 0, Time1: 0	
SigB3	Source: „0”, Mode: 2, Negation: 1 Time: 0, Time0: 0, Time1: 0	
SigC0	Source: „0”, Time: 0	
SigC1	Source: „0”, Time: 0	
SigC2	Source: „0”, Time: 0	
SigC3	Source: „0”, Time: 0	
<b>Hasło</b>		
Hasło	„”, 0x3C	brak hasła, 60s

## 10 BOOTLOADER - ZMIANA FIRMWARE URZĄDZENIA

### 10.1 AKTUALIZACJA Z WYKORZYSTANIEM APLIKACJI NEFIR 1.9.1

W celu wgrania do urządzenia nowego firmware, należy postępować według poniższej procedury:

1. Podłączyć urządzenie do portu szeregowego RS232 w komputerze
2. Otworzyć program NEFIR.exe
3. Ustawić odpowiedni port COM oraz prędkość transmisji na 9600bps
4. Wcisnąć przycisk *Open HEX File* i wczytać plik z nowym firmware
5. Wcisnąć przycisk *START*, który spowoduje rozpoczęcie przeładowywania firmware
6. Poczekać na zakończenie procesu przeładowywania

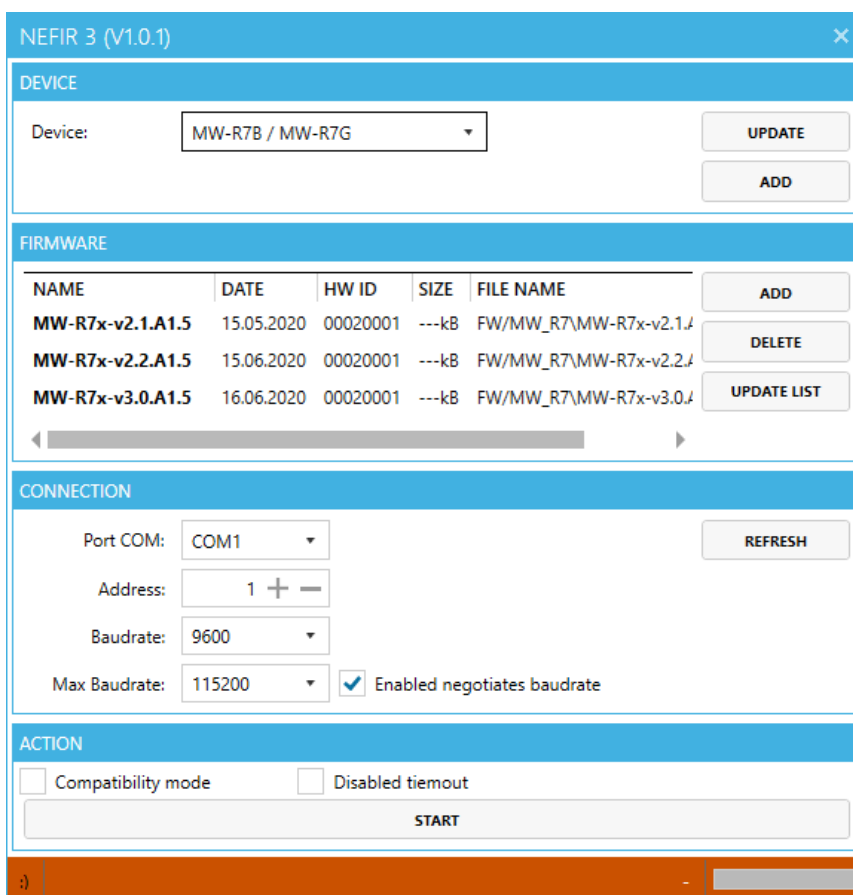


Rysunek 10.1 Widok okna programu podczas przeładowywania firmware

## 10.2 AKTUALIZACJA Z WYKORZYSTANIEM APLIKACJI NEFIR 3

W celu wgrania do urządzenia nowego firmware, należy postępować według poniższej procedury:

1. Podłączyć urządzenie do portu szeregowego RS232 w komputerze
2. Otworzyć program NEFIR3.exe
3. W sekcji *DEVICE* z rozwijanego pola *Device* wybrać *MW-R7B / MW-R7G*. Jeżeli nie ma takiego urządzenia na liście, to należy kliknąć przycisk *UPDATE*.
4. W sekcji *FIRMWARE* zaznaczyć wersję firmware, którą chcemy zaktualizować urządzenie. Listę dostępnych wersji firmware można zaktualizować klikając na przycisk *UPDATE*.
5. W sekcji *CONNECTION*:
  - a. Ustawić odpowiedni port COM
  - b. Ustawić adres urządzenia (standardowo 0x01).
  - c. Ustawić prędkość transmisji (*Baudrate*), na jakiej pracował do tej pory czytnik (standardowo 9600bps).
  - d. Opcjonalnie zaznaczyć pole wyboru *Enabled negotiates baudrate* i ustawić maksymalną prędkość transmisji (*Max. Baudrate*) przy aktualizacji oprogramowania. Skorzystanie z tej opcji znacząco przyspiesza proces aktualizacji.
6. Wcisnąć przycisk *START*, który spowoduje rozpoczęcie przeładowywania firmware
7. Poczekać na zakończenie procesu przeładowywania



Rysunek 10.2 Widok okna programu NEFIR3 podczas przeładowywania firmware