

# Instrukcja obsługi

---

## Czytnik RFID UHF dalekiego zasięgu

### HD-RD06-UHF

|  |    |
|--|----|
| Spis treści  |    |
| Specyfikacje: .....  | 5  |
| Zawartość zestawu: .....   | 6  |
| Opis protokołu .....   | 6  |
| Formaty ramek z danymi .....   | 7  |
| Format ramki poleceń (od HOST).....                                    | 7  |
| Format ramki odpowiedzi (od CZYTNIKA) .....                            | 9  |
| Podsumowanie poleceń .....   | 11 |
| Komendy 18000-6B .....   | 12 |
| Polecenie niestandardowe czytnika.....                                 | 14 |
| Kody błędu tagu RFID.....  | 17 |
| Pamięć znaczników i instrukcje dostępu.....                            | 19 |
| Szczegółowy opis polecenia.....  | 20 |
| Polecenia EPC C1G2 .....   | 21 |
| Tagi zapasów .....   | 21 |
| Odczyt danych.....   | 30 |
| Zapis danych .....   | 33 |
| Wprowadzanie numeru EPC.....   | 36 |
| Ustawienie ochrony przed odczytem/zapisem dla określonej pamięci ..... | 37 |
| Wymazywanie bloków.....  | 41 |
| Odczyt konfiguracji ochrony (zgodnie z numerem EPC) .....              | 44 |
| Odczyt konfiguracji ochrony (bez numeru EPC).....                      | 47 |
| Odblokuj ochronę przed odczytem.....                                   | 48 |
| Odczytywanie sprawdzania stanu ochrony.....                            | 49 |
| Konfiguracja EAS .....   | 50 |
| Wykrywanie alertów EAS.....  | 52 |
| Zasoby reklamowe z pojedynczym tagiem .....                            | 53 |
| Blokuj polecenie pisania.....  | 54 |
| Uzyskanie parametrów pracy Monza4QT .....                              | 57 |
| Zmodyfikuj parametry pracy Monza4QT .....                              | 59 |
| Rozszerzony odczyt danych z przypisaną maską .....                     | 62 |
| Rozszerzony odczyt danych z przypisaną maską .....                     | 65 |
| Inwentaryzacja z buforem pamięci .....                                 | 68 |
| Mieszanie zasobów reklamowych .....                                    | 3  |

|   |    |
|---|----|
| Inwentaryzacja z numerem EPC .....  | 11 |
| Zapasy QT .....   | 14 |
| Polecenia 18000-6B .....  | 19 |
| Zapytaj o tagi zgodnie z określonym warunkiem.....  | 19 |
| Dane czytnika .....   | 22 |
| Zapis danych .....  | 23 |
| Uzyskiwanie stanu blokady.....  | 24 |
| Bajt blokady .....  | 24 |
| Indywidualne polecenia czytnika .....   | 25 |
| Zmodyfikuj częstotliwość roboczą .....  | 28 |
| Zmień adres czytnika .....  | 31 |
| Modyfikowanie czasu inwentaryzacji czytnika.....  | 31 |
| Modyfikowanie szybkości transmisji szeregowej .....   | 32 |
| Zmodyfikuj moc RF.....  | 34 |
| Sterowanie brzęczykiem/diodą LED .....  | 34 |
| Konfiguracja multipleksowania anten.....  | 35 |
| Włącz/wyłącz brzęczyk.....  | 36 |
| Sterowanie GPIO .....   | 37 |
| Uzyskiwanie stanu GPIO .....  | 38 |
| Uzyskaj unikalny numer seryjny czytnika .....   | 38 |
| Modyfikowanie funkcji niestandardowej tagu .....  | 39 |
| Włącz sprawdzanie anteny.....   | 40 |
| Modyfikowanie interfejsu komunikacyjnego .....  | 40 |
| Modyfikowanie lub wczytywanie konfiguracji progu tłumienia odbicia anteny .....                 | 41 |
| Modyfikowanie konfiguracji maksymalnej długości EPC/TID dla bufora pamięci .....                | 43 |
| Łaładuj konfigurację maksymalnej długości EPC/TID .....   | 43 |
| Uzyskiwanie danych z bufora pamięci .....   | 44 |
| Wyczyść bufor pamięci.....  | 46 |
| Uzyskaj łączną ilość tagów z bufora pamięci.....  | 46 |
| Modyfikacja parametrów trybu inwentaryzacji w czasie rzeczywistym.....                          | 46 |
| Modyfikowanie trybu pracy czytnika .....  | 50 |
| Uzyskaj tryb pracy czytnika.....  | 53 |
| Łaładuj/zmodyfikuj przerwę w czasie pakietów pulsu w inwentaryzacji w czasie rzeczywistym ..... | 55 |
| Modyfikowanie konfiguracji mocy RF oddzielnie dla operacji zapisu .....                         | 57 |

|  |    |
|--|----|
| Załaduj konfigurację mocy RF operacji zapisu .....                               | 58 |
| Modyfikowanie lub ładowanie konfiguracji maksymalnego czasu ponawiania zapisu .. | 58 |
| Modyfikowanie hasła funkcji niestandardowych tagów .....                         | 60 |
| Uzyskaj hasło do funkcji niestandardowych tagów .....                            | 61 |
| Załaduj/zmodyfikuj profil czytnika.....  | 61 |
| Synchronizuj znacznik czasu EM4325 .....   | 63 |
| Uzyskaj dane o temperaturze EM4325.....  | 64 |
| Uzyskiwanie danych zewnętrznych za pośrednictwem EM4325 SPI .....                | 68 |
| Zresetuj alert EM4325.....   | 72 |
| Modyfikowanie lub wczytywanie konfiguracji DRM .....                             | 74 |
| Zmierz tłumienie odbicia anteny .....  | 75 |
| Zmierz aktualną temperaturę czytnika.....  | 77 |
| Ustawianie parametrów rozszerzonych .....  | 77 |
| Ustawianie parametrów rozszerzonych .....  | 78 |
| Rozpocznij szybką inwentaryzację .....   | 79 |
| Zatrzymaj szybką inwentaryzację.....   | 80 |
| Indywidualne polecenia czytnika .....  | 80 |
| Tabela parametrów konfiguracyjnych czytnika.....                                 | 81 |
| Opis funkcji parametru czytnika .....  | 81 |

## Specyfikacje:

- **Gwarancja:** 1 rok
- **Kolor:** biały
- **Materiał wykonania:** ABS, żeliwo
- **Częstotliwość pracy:** 865–868 MHz, 902–928 MHz
- **RFID wyjściowe:** od 0 dBm do 26dBm\* z +/-1dBm
- **Odległość odczytu:** do 2 metrów
- **Zintegrowana antena:** 5,5 dBm
- **Szczytowa prędkość inwentaryzacji:**  
>50 tagów/sekundę
- **RSSI tagu:** obsługiwany
- **Detektor antenowy:** obsługiwany
- **Interfejs komunikacyjny:** 10M/100M Adaptacyjny Ethernet, RS232
- **Gniazda:** DC, GPIO, RS232, RJ45
- **Zasilanie:** 220 V AC, wyjście konwertera zasilania +12 V/5A DC
- **System operacyjny:** Linux
- **Wymiary urządzenia:** 12,9 x 12,9 x 4,9 cm
- **Wymiary produktu:** 25,5 x 20 x 7 cm
- **Waga urządzenia:** 450 g
- **Waga z opakowaniem:** 1250 g
- **Temperatura pracy:** -20°C do 55°C
- **Temperatura przechowywania:** -20°C do +85°C
- **Wilgotność pracy:** 5% ~ 95% bez kondensacji
- **Wilgotność przechowywania:** 5% ~ 95% bez kondensacji

## Zawartość zestawu:

- Czytnik tagów RFID
- Zasilacz
- Śruby montażowe
- Metalowa płytka

## Opis protokołu

W typowej komunikacji, host dostarcza komendę i parametry do czytnika, czytnik powinien wykonać komendę i dostarczyć wynik lub odpowiedź danych. Czytnik otrzyma i wykona JEDNĄ komendę w każdym cyklu komunikacji. Czytnik nie będzie w stanie odebrać nowego polecenia przed zakończeniem wykonywania ostatniego i dostarczeniem odpowiedzi. Każda komenda przesłana podczas wykonywania komendy zostanie zignorowana przez czytnik.

### Dostarczenie polecenia z hosta:

| Host            | Kierunek | Czytnik |
|-----------------|----------|---------|
| Ramka polecenia | ->       |         |
|                 |          |         |

### Opis:

Po odebraniu polecenia przez czytnik, urządzenie wygeneruje odpowiedź do hosta w ciągu predefiniowanego czasu inwentaryzacji. Czas transmisji danych jest tutaj nieistotny.

Podczas komunikacji, przerwa w transmisji pomiędzy każdymi dwoma kolejnymi bajtami powinna być mniejsza niż 15ms. Jeśli przerwa przekracza 15 ms, wszystkie odebrane dane powinny zostać odrzucone, a normalny proces odbioru rozpocznie się od następnego bajtu.

### Dostarczenie odpowiedzi przez czytnik:



| Czytnik         | Kierunek | Host |
|-----------------|----------|------|
| Rama odpowiedzi | ->       |      |

Opis:

Podczas komunikacji przerwa w nadawaniu między dwoma kolejnymi bajtami powinna być mniejsza niż 15 ms.

Poniżej przedstawiono typowy proces komunikacji:

1. Host wysyła polecenie do czytnika i czeka na odpowiedź;
2. Czytnik odbiera polecenie, rozpoczyna określoną operację zgodnie z odebraną ramką i generuje odpowiedź (statusu operacji i/lub danych) do hosta;
3. Host odbiera odpowiedź od czytnika

## Formaty ramek z danymi

Format ramki poleceń (od HOST)

| Len | Adr | Cmd | Data[] | LSB-<br>CRC16 | MSB-<br>CRC16 |
|-----|-----|-----|--------|---------------|---------------|
|     |     |     |        |               |               |

Opis formatów:

|     | Długość (w bajtach) | Opis   |
|-----|---------------------|--|
| Len | 1                   | Długość ramki poleceń, określona przez liczbę bajtów między Adr i MSB_CRC16, tj. długość |

|           |         |   |
|-----------|---------|---|
|           |         | Data[] + 4.<br>Minimalne i<br>maksymalne<br>wartości Len to 4<br>i 255.   |
| Adr       | 1       | Adres urządzenia<br>czytnika,<br>prawidłowy zakres<br>adresu urządzenia<br>to<br>0x00 ~ 0xFE. 0xFF to<br>adres<br>rozgłoszeniowy.<br>Czytnik odpowie<br>tylko na ramkę<br>polecenia podaną z<br>adresem<br>identycznym lub<br>0xFF.<br>Domyślny adres<br>urządzenia: 0x |
| Cmd       | 1       | Kod polecenia   |
| Data[]    | Zmienna | Pole parametrów,<br>host może, ale nie<br>musi dostarczać<br>parametrów<br>zgodnie z różnymi<br>formatami poleceń.  |
| LSB-CRC16 | 1       | Niski bajt CRC16.<br>Te 2 bajty są<br>obliczeniami CRC16<br>od Len do Data[]  |

|           |   |                   |
|-----------|---|-------------------|
| MSB-CRC16 | 1 | Wysoki bajt CRC16 |
|-----------|---|-------------------|

Format ramki odpowiedzi (od CZYTNIKA)

|     |     |       |        |        |           |           |
|-----|-----|-------|--------|--------|-----------|-----------|
| Len | Adr | reCmd | Status | Data[] | LSB-CRC16 | MSB-CRC16 |
|-----|-----|-------|--------|--------|-----------|-----------|

Opis formatów:

|        | Długość (w bajtach) | Opis   |
|--------|---------------------|--|
| Len    | 1                   | Długość ramki odpowiedzi, określona przez liczbę bajtów między Adr i MSB_CRC16), tj. długość Data[] + 5.           |
| Adr    | 1                   | Adres czytnika   |
| reCmd  | 1                   | reCmd wskazuje, na które polecenie odpowiada ta ramka<br>. Dla nierozpoznanej komendy, reCmd powinno wynosić 0x00. |
| Status | 1                   | Status/wynik wykonania polecenia   |
| Data[] | Zmienna             | Pole danych, czytnik może, ale nie musi  |

|           |   |  |
|-----------|---|--|
|           |   | dostarczać parametrów zgodnie z rzeczywistymi poleceniami                    |
| LSB-CRC16 | 1 | Niski bajt CRC16. These 2 bytes are the CRC16 calculation of Len to Data[.]. |
| MSB-CRC16 | 1 | Wysoki bajt CRC16  |

Algorytm CRC16 w języku C:

```
#define PRESET_VALUE 0xFFFF #define POLYNOMIAL 0x8408
unsigned int uiCrc16Cal(unsigned char const * pucY, unsigned char
ucX)
{
unsigned char ucl,ucJ;
unsigned short int uiCrcValue = PRESET_VALUE; for(ucl = 0; ucl < ucX;
ucl++)
{
uiCrcValue = uiCrcValue ^ *(pucY + ucl); for(ucJ = 0; ucJ < 8; ucJ++)
{
if(uiCrcValue & 0x0001) {
uiCrcValue = (uiCrcValue >> 1) ^ POLYNOMIAL;
}e lse {
```

```

uiCrcValue = (uiCrcValue >> 1); }
} }r eturn uiCrcValue; }

```

`pucY` jest wskaźnikiem wpisu tablicy do obliczenia CRC16, `ucX` jest ilością bajtów uwzględnionych w tym obliczeniu. Wartość zwracana `0x0000` wskazuje, że dane wejściowe przeszły weryfikację CRC16. Za każdym razem, gdy host otrzymuje ramkę danych z czytnika, host powinien zawsze zweryfikować ramkę przez wywołanie tego algorytmu CRC16.

## Podsumowanie poleceń

| Lp. | Kod polecenia | Funkcja  |
|-----|---------------|--|
| 1   | 0x01          | Spis tagów   |
| 2   | 0x02          | Polecenie odczytu danych   |
| 3   | 0x03          | Polecenie zapisu danych  |
| 4   | 0x04          | Numer EPC  |
| 5   | 0x05          | Kill tag   |
| 6   | 0x06          | Ustawienie ochrony pamięci przed odczytem/zapisem dla określonej pamięci |
| 7   | 0x07          | Wymazywanie bloku  |
| 8   | 0x08          | Konfiguracja ochrony przed odczytem (zgodnie z numerem EPC)              |
| 9   | 0x09          | Konfiguracja ochrony przed odczytem (bez numeru EPC)                     |
| 10  | 0x0a          | Odblokowanie ochrony przed odczytem                                      |

|    |      |   |
|----|------|---|
| 11 | 0x0b | Sprawdzanie stanu ochrony przed odczytem        |
| 12 | 0x0c | Konfiguracja EAS                                |
| 13 | 0x0d | Wykrywanie alarmów EAS                          |
| 14 | 0x0f | Inwentaryzacja za pomocą pojedynczego znacznika |
| 15 | 0x10 | Bloki zapisu                                    |
| 16 | 0x11 | Uzyskanie parametrów roboczych Monza4QT         |
| 17 | 0x12 | Modyfikacja parametrów pracy Monza4QT           |
| 18 | 0x15 | Rozszerzony odczyt danych z przypisaną maską    |
| 19 | 0x16 | Rozszerzony zapis danych z przypisaną maską     |
| 20 | 0x18 | Inwentaryzacja z buforem pamięci                |
| 21 | 0x19 | Mix zapasów                                     |
| 22 | 0x1a | Inwentaryzacja z numerem EPC                    |
| 23 | 0x1b | Inwentaryzacja QT                               |

### Komendy 18000-6B

| Lp. | Kod polecenia | Funkcja  |
|-----|---------------|--|
| 1   | 0x50          | Polecenie inwentaryzacji dla POJEDYNCZEGO tagu. To polecenie zapyta tylko o jeden znacznik w każdym cyklu, bez |

|   |      |   |
|---|------|---|
|   |      | warunku inwentaryzacji.   |
| 2 | 0x51 | Polecenie inwentaryzacji dla WIELU tagów.<br>Zapytanie o tagi zgodnie z predefiniowanym warunkiem i zwrócenie tylko identyfikatorów UID odpowiednich tagów. |
| 3 | 0x52 | Polecenie odczytu danych.<br>Odczyt danych z tagu, maksymalnie 32 bajty w każdym cyklu komendy.   |
| 4 | 0x53 | Polecenie zapisu danych.<br>Zapis danych do tagu, maksymalnie 32 bajty w każdym cyklu komendy.  |
| 5 | 0x54 | Polecenie uzyskania stanu blokady.<br>Sprawdza stan blokady określonej jednostki pamięci.   |
| 6 | 0x55 | Polecenie blokowania bajtów.<br>Blokuje określony (odblokowany) bajt w tagu.  |

## Polecenie niestandardowe czytnika

| Lp. | Kod polecenia | Funkcja  |
|-----|---------------|--|
| 1   | 0x21          | Uzyskiwanie informacji o czytniku                                      |
| 2   | 0x22          | Modyfikacja częstotliwości pracy                                       |
| 3   | 0x24          | Modyfikacja adresu czytnika  |
| 4   | 0x25          | Modyfikacja czasu inwentaryzacji czytnika                              |
| 5   | 0x28          | Modyfikacja szybkości transmisji szeregowej                            |
| 6   | 0x2f          | Modyfikacja mocy RF  |
| 7   | 0x33          | Sterowanie diodami LED/brzęczykiem                                     |
| 8   | 0x3f          | Konfiguracja multipleksowania anteny                                   |
| 9   | 0x40          | Włączanie/wyłączanie brzęczyka   |
| 10  | 0x46          | Sterowanie GPIO  |
| 11  | 0x47          | Uzyskanie stanu GPIO   |
| 12  | 0x4c          | Uzyskaj unikalny numer seryjny czytnika                                |
| 13  | 0x3a          | Zmodyfikuj niestandardową funkcję tagu                                 |
| 14  | 0x66          | Włącz sprawdzanie anteny   |
| 15  | 0x6a          | Modyfikacja interfejsu komunikacyjnego                                 |
| 16  | 0x6e          | Modyfikacja lub załadowanie konfiguracji progu strat powrotnych anteny |

|    |      |  |
|----|------|--|
| 17 | 0x70 | Modyfikacja konfiguracji maksymalnej długości EPC/TID dla bufora pamięci                 |
| 18 | 0x71 | Załaduj konfigurację maksymalnej długości EPC/TID  |
| 19 | 0x72 | Uzyskanie danych z bufora pamięci  |
| 20 | 0x73 | Wyczyść bufor pamięci  |
| 21 | 0x74 | Uzyskanie całkowitej ilości tagów z bufora pamięci                                       |
| 22 | 0x75 | Modyfikacja parametrów trybu inwentaryzacji w czasie rzeczywistym                        |
| 23 | 0x76 | Modyfikacja trybu pracy  |
| 24 | 0x77 | Ładowanie parametrów trybu inwentaryzacji w czasie rzeczywistym                          |
| 25 | 0x78 | Załaduj/modyfikuj przerwę czasową pakietu heartbeat inwentaryzacji w czasie rzeczywistym |
| 26 | 0x79 | Modyfikacja konfiguracji mocy RF oddzielnie dla operacji zapisu                          |
| 27 | 0x7a | Załaduj konfigurację mocy RF operacji zapisu   |
| 28 | 0x7b | Modyfikowanie lub wczytywanie  |

|    |      |  |
|----|------|--|
|    |      | konfiguracji<br>maksymalnego czasu<br>ponawiania zapisu      |
| 29 | 0x7d | Modyfikacja hasła<br>funkcji<br>niestandardowych<br>tagów    |
| 30 | 0x7e | Uzyskanie hasła do<br>niestandardowych<br>funkcji tagu       |
| 31 | 0x7f | Załaduj/modyfikuj<br>profil czytnika                         |
| 32 | 0x85 | Synchronizacja<br>znacznika czasu<br>EM4325                  |
| 33 | 0x86 | Uzyskanie danych<br>temperatury EM4325                       |
| 34 | 0x87 | Uzyskiwanie danych<br>zewnętrznych przez<br>EM4325 SPI       |
| 35 | 0x88 | Resetowanie alarmu<br>EM4325                                 |
| 36 | 0x90 | Modyfikowanie lub<br>ładowanie konfiguracji<br>DRM           |
| 37 | 0x91 | Pomiar strat<br>odbiciowych anteny                           |
| 38 | 0x92 | Pomiar aktualnej<br>temperatury czytnika                     |
| 39 | 0x50 | Szybkie rozpoczęcie<br>inventaryzacji (tylko<br>seria Ex10)  |
| 40 | 0x51 | Zatrzymanie szybkiej<br>inventaryzacji (tylko<br>seria Ex10) |
| 41 | 0xEA | Ustawianie<br>parametrów                                     |

|    |      |  |
|----|------|--|
|    |      | rozszerzonych<br>(obsługuje tylko serię Ex10)              |
| 42 | 0xEB | Pobierz rozszerzone parametry (obsługuje tylko serię Ex10) |

## Kody błędów tagu RFID

| Typ kodu błędu      | Kod błędu                              | Definicja kodu błędu  | Opis błędu                                 |
|---------------------|--|---|--|
| Określony kod błędu | 0x00                                   | Inne błędy  | Wszystkie inne błędy obsługiwane przez tag |
| 0x03                | Pamięć pełna lub nielegalna wartość PC | Lokalizacja pamięci nie została znaleziona lub nieobsługiwana wartość PC. |  |
| 0x04                | Zablokowana pamięć                     | Pamięć jest tymczasowo / trwale zablokowana, nie można wykonać            |  |

|      |                            |   |  |
|------|----------------------------|---|--|
|      |                            | operacji zapisu.  |  |
| 0x0b | Niewystarczające zasilanie | Nie można wykonać operacji zapisu z powodu niewystarczającego zasilania tagu. |  |

| Typ kodu błędu      | Kod błędu                                       | Definicja kodu błędu  | Opis błędu                                  |
|---------------------|---|---|---|
| Określony kod błędu | 0x00  | Inne błędy  | Wszystkie inne błędy obsługiwane przez tag. |
| 0x03                | Pełna pamięć lub niedozwolona wartość komputera | Lokalizacja pamięci nie została znaleziona lub nieobsługiwana wartość PC.       |   |
| 0x04                | Pamięć zablokowana                              | Pamięć jest tymczasowo / trwale zablokowana, nie można wykonać operacji zapisu. |   |
| 0x0b                | Niewystarczający zasilacz                       | Nie można wykonać operacji zapisu z   |   |

|                        |      |   |                                 |
|------------------------|------|---|---------------------------------|
|                        |      | powodu niewystarczającego zasilania do tagowania. |                                 |
| Nieokreślony kod błędu | 0x0f | Niedefiniowany błąd                               | Oznaczone nieobsługiwane błędy. |

## Pamięć znaczników i instrukcje dostępu

Istnieją 4 obszary pamięci w tagach G2, pamięć zarezerwowana (znana również jako pamięć hasła), pamięć EPC, pamięć EID i pamięć użytkownika.

**Pamięć zarezerwowana:** 4 bajty, pierwsze 2 bajty to hasło zabicia tagu, a pozostałe 2 bajty to hasło dostępu. Pamięć ta jest domyślnie dostępna do odczytu i zapisu. Konfiguracje ochrony przed odczytem/zapisem dwóch bloków haseł są odizolowane i mogą być modyfikowane oddzielnie.

**Pamięć EPC:** numer EPC tagu jest przechowywany w tej pamięci. Słowo 0 to CRC16 wartości PC i numer EPC. Word1 to wartość PC, wskazuje długość numeru EPC. Rzeczywisty numer EPC to z Word2. Pamięć ta jest domyślnie dostępna do odczytu i zapisu.

**Pamięć TID:** w tej pamięci przechowywany jest wstępnie zdefiniowany przez producenta identyfikator tagu. Jest ona domyślnie odczytywalna przez, ale niezapisywalna.

**Pamięć użytkownika:** w tej pamięci przechowywane są dane użytkownika. Domyślnie można ją odczytywać i zapisywać.

**UWAGA:**

- Długość danych jest wymagana w wielu poleceniach G2, jednostką długości danych jest słowo lub bajt. Długość 1 słowa jest równa długości 2 bajtów.
- Niektóre polecenia mogą wymagać hasła dostępu, jeśli hasło dostępu nie jest ustawione w tagu docelowym, wypełnij pole hasła wartością 0. Pole hasła nie może być puste.

## Szczegółowy opis polecenia

Istnieją 2 rodzaje poleceń:

- polecenia związane z protokołem;
- polecenia związane z czytnikiem;

Jeśli nieprawidłowa ramka polecenia zostanie dostarczona z hosta, np. nierozpoznany kod polecenia lub ramka polecenia z błędem CRC (nie przeszła kontroli CRC16). Czytnik odpowie podobną ramką, jak pokazano poniżej.

| Len  | Adr  | reCmd | Status | CRC-16 |      |
|------|------|-------|--------|--------|------|
| 0x05 | 0xXX | 0x00  | 0xFE   | LSB    | 0x05 |

W przypadku nieprawidłowej długości ramki polecenia odpowiedź będzie podobna do następującej ramki

| Len  | Adr  | reCmd | Status | CRC-16 |     |
|------|------|-------|--------|--------|-----|
| 0x05 | 0xXX | 0xXX  | 0xFD   | LSB    | MSB |

Istnieją 2 sytuacje, w których komendy będą ignorowane przez czytnik:

- Nieprawidłowy adres (ani adres nadawczy 0xff, ani odpowiadający mu adres czytnika)
- Niekompletna ramka komendy, tj. rzeczywista długość ramki komendy jest mniejsza niż wartość wskazana w polu "Len"  
Czytnik nie udzieli żadnej odpowiedzi w sytuacjach opisanych powyżej.

## Polecenia EPC C1G2

### Tagi zapasów

Komenda inwentaryzacji jest używana do zapytania tagów RFID z odpowiednim protokołem i uzyskania informacji o tagu w efektywnym polu. Aby przetworzyć dalsze operacje na nowym tagu z nieznanym EPC, użytkownik powinien najpierw uzyskać numer EPC tagu za pomocą polecenia inwentaryzacji.

W zależności od okoliczności, użytkownik może zmodyfikować ustawienie maksymalnego czasu pracy (czas inwentaryzacji). Czytnik zwróci wynik inwentaryzacji w określonym czasie. Jeśli czytnik nie zakończy procesu inwentaryzacji wszystkich tagów w efektywnym polu w zdefiniowanym czasie inwentaryzacji, czytnik nie przeprowadzi dalszej inwentaryzacji po przekroczeniu czasu inwentaryzacji. Czytnik przekaże do hosta informacje o wszystkich już sprawdzonych tagach, a także o bieżącym (nieukończonym) stanie inwentaryzacji. Następnie czytnik będzie oczekiwał na kolejną komendę od hosta.

Domyślny czas inwentaryzacji wynosi 2 s, można go zmienić za pomocą niestandardowej komendy czytnika "modyfikuj czas inwentaryzacji czytnika". Prawidłowy zakres czasu inwentaryzacji wynosi od 3\*100ms do 255\*100ms (rzeczywisty czas odpowiedzi może być nieco większy niż ustawiona wartość, a różnica będzie się wahać od 0 do 75ms).

**Uwaga:** ustawienie zbyt krótkiego czasu inwentaryzacji może spowodować niepowodzenie inwentaryzacji tagu.

Ramka poleceń:

| Len  | Adr  | Cmd  | Data[] | CRC-16 |     |
|------|------|------|--------|--------|-----|
| 0xXX | 0xXX | 0x01 | —      | LSB    | MSB |

Parametry Data[]:

| Data[] |         |         |         |         |          |        |        |
|--------|---------|---------|---------|---------|----------|--------|--------|
| QValue | Session | MaskMem | MaskAdr | MaskLen | MaskData | AdrTID | LenTID |
| 0xXX   | 0xXX    | 0xXX    | 2 bytes | 0xXX    | Variable | 0xXX   | 0xXX   |

| Data[] |      |          |
|--------|------|----------|
| Target | Ant  | Scantime |
| 0xXX   | 0xXX | 0xXX     |

## Definicja parametrów

QValue: 1 bajt.

bit7: flaga pakietu danych statystycznych.

0 – po inwentaryzacji, NIE dostarczaj pakietu danych statystycznych procesu inwentaryzacji;

1 – po inwentaryzacji, dostarcz pakiet danych statystycznych procesu inwentaryzacji

bit6: wskaźnik strategii.

0 – strategia ogólna;

1 – strategia specjalna.

bit5: wskaźnik inwentaryzacji FastID.

0 – wyłączenie funkcji Impinj FastID;

1 – włączenie funkcji Impinj FastID.

Bit4: Informacja o fazie.

0 – wyłącz;

1 – włącz.

Bit3 ~ bit0: oryginalna wartość Q inwentaryzacji tagu EPC, ustawienie wartości Q powinno być zgodne z regułą:  $2Q \approx$  całkowita ilość tagów w efektywnym polu. Zakres wartości Q wynosi 0 ~ 15,, jeśli w tym polu zostanie podana inna wartość, czytnik zwróci status błędu parametru w ramce odpowiedzi.

Session: 1 bajt, wartość sesji inwentaryzacji znacznika EPC.

0x00 – zastosuj S0 jako wartość sesji;

0x01 – zastosuj S1 jako wartość sesji;

0x02 – zastosuj S2 jako wartość sesji;

0x03 – zastosuj S3 jako wartość sesji;

0xff - zastosuj inteligentną konfigurację czytnika (ważne tylko w inwentarzu EPC).

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli inna wartość zostanie dostarczona w tym polu.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 - pamięć EPC;

0x02 - pamięć TID;

0x03 - pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli inna wartość zostanie podana w tym polu.

**MaskAdr:** 2 bajty, bitowy adres wejściowy maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące powinny być wypełnione 0.

**AdrTID:** adres wejściowy pamięci TID inwentaryzacji. Jeśli komenda przekazała ten parametr i LenTID w tej samej ramce, trybem inwentaryzacji powinien być TID, a nie EPC.

**LenTID:** długość danych dla operacji inwentaryzacji TID, prawidłowy zakres LenTID to 0 ~ 15. Wszystkie inne wartości są zarezerwowane, a czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli inna wartość zostanie dostarczona w tym polu. Jeśli polecenie

dostarczyło ten parametr i AdrTID, trybem inwentaryzacji powinien być TID, a nie EPC.

**Target (parametr opcjonalny):** 1 bajt, wartość docelowa inwentaryzacji znacznika EPC.

0x00 – zastosuj A jako wartość docelową;

0x01 – zastosuj B jako wartość docelową.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli inna wartość zostanie podana w tym polu.

**Ant (parametr opcjonalny):** 1 bajt, wybór anteny dla bieżącego spisu.

0x80 – antena 1;

0x81 – antena 2;

0x82 – antena 3;

0x83 – antena 4.

Wszystkie inne wartości są zarezerwowane, a czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli inna wartość zostanie podana w tym polu.

**ScanTime (parametr opcjonalny):** 1 bajt, konfiguracja czasu inwentaryzacji dla bieżącej inwentaryzacji. Czytnik ustawi maksymalny czas działania jako  $\text{ScanTime} \times 100\text{ms}$ .

#### **UWAGA:**

1. Jeśli parametry MaskMem, MaskAdr, MaskLen i MaskData nie zostaną podane, żaden wzorzec maski nie będzie zaangażowany w inwentaryzację

2. Target, Ant i ScanTime są parametrami opcjonalnymi, wszystkie trzy parametry muszą być dostarczone w jednej ramce polecenia. Czytnik zastosuje domyślne ustawienia dla tych parametrów, dopóki wszystkie trzy parametry nie zostaną prawidłowo dostarczone z hosta, nowe ustawienia staną się skuteczne dla po inwentaryzacji.

3. Funkcja FastID jest ważna tylko dla wkładki obsługującej FastID od Impinj.

(1) Status odpowiedzi to 0x26, odpowiedź powinna wyglądać następująco

| Len  | Adr  | reCmd | Status | Data[] |          |            | CRC-16 |     |
|------|------|-------|--------|--------|----------|------------|--------|-----|
|      |      |       |        | Ant    | ReadRate | TotalCount |        |     |
| 0xXX | 0xXX | 0x01  | 0x26   | 0xXX   | 2 bytes  | 4 bytes    | LSB    | MSB |

## Definicja parametrów

Status: status odpowiedzi, szczegółowy opis przedstawiono w poniższej tabeli

| Status | Opis   |
|--------|--|
| 0x26   | Po zakończeniu inwentaryzacji dostarcz dane statystyczne z ostatniego procesu inwentaryzacji |

**Ant:** wskazuje, która antena zapytała o tag. Na przykład, 0x04 to 0000 0100 w binarnym, to wskazuje, że Antena 3 zapytała o ten konkretny tag. Dla 0x08, jest to 0000 1000 w binarnym, to wskazuje, że Antena 4 zapytała o informacje o tagu.

**ReadRate:** wskaźnik identyfikacji tagu dla bieżącego spisu (czas udanej identyfikacji / sek.), tag identyfikowany wielokrotnie również wydłuży czas udanej identyfikacji.

**TotalCount:** całkowita ilość tagów wykrytych w bieżącym spisie, tag, do którego uzyskano dostęp wiele razy, również zwiększy całkowitą ilość.

(2) Status odpowiedzi nie jest 0x26, odpowiedź powinna być następująca:

| Len  | Adr  | reCmd | Status | Data[] |      |                        | CRC-16 |     |
|------|------|-------|--------|--------|------|------------------------|--------|-----|
|      |      |       |        | Ant    | Num  | EPC ID                 |        |     |
| 0xXX | 0xXX | 0x01  | 0xXX   | 0xXX   | 0xXX | EPC-1, EPC-2, EPC-3... | LSB    | MSB |

## Definicja parametrów

Status: status odpowiedzi, szczegółowy opis przedstawiono w poniższej tabeli

| Status | Opis  |
|--------|---|
| 0x01   | Operacja zakończona, czytnik zgłosi wszystkie wymagane informacje o tagach do hosta                           |
| 0x02   | Limit czasu inwentaryzacji, operacja zostanie przerwana. czytnik zgłosi wszystkie już zapytane tagi do hosta. |

|      |  |
|------|--|
| 0x03 | Dla statusu = 0x03, czytnik nie jest w stanie odpowiedzieć na wszystkie dane w pojedynczej ramce, dalsze dane będą przesyłane w kolejnych ramkach.   |
| 0x04 | Czytnik wykonał tylko część inwentaryzacji, ale zabrakło mu miejsca w pamięci ze względu na ilość tagów. Czytnik zgłosi wszystkie już sprawdzone tagi jako, a także tę wartość statusu do hosta. |
| 0xF8 | Wykryto błąd anteny, bieżąca antena może być odłączona.  |

**Ant:** wskazuje, która antena zapytała o tag. Na przykład, 0x04 to 0000 0100 w binarnym, to wskazuje, że Antena 3 zapytała o ten konkretny tag. Dla 0x08, jest to 0000 1000 w binarnym, to wskazuje, że Antena 4 zapytała o informacje o tagu.

**Num:** ilość RFID EPC/TID zawarta w tej odpowiedzi.

**EPC ID:** zapytane dane EPC/TID. Na przykład EPC-1 to długość EPC/TID + numer EPC/TID dane + wartość RSSI pierwszego tagu itp. Wartości RSSI i długości EPC są parametrami jednobajtowymi. Jednostką danych EPC C1 G2 jest słowo. Podczas komunikacji najbardziej znaczące słowo i najbardziej znaczący bajt w słowie zostaną przesłane jako pierwsze.

W sytuacji, gdy funkcja FastID jest włączona, każdy blok danych EPC ID jest zgodny z tym formatem:

Całkowita długość EPC plus TID (1 bajt) + numer EPC znacznika + 12 bajtów danych TID + wartość RSSI znacznika

Najbardziej znaczący bit całkowitej długości EPC i TID wskazuje, czy TID jest zawarty w tym bloku.

## Szczegółowa struktura EPC ID

| EPC ID   |         |       |        |                     |
|--|---------|-------|--------|---------------------|
| EPC-1  |         |       |        |                     |
| Długość danych   | Dane    | RSSI  | faza   | Częstotliwość (kHz) |
| 1 bit  | N bitów | 1 bit | 4 bity | 3 bity              |
| bit7:<br>0 - blok danych zawiera EPC lub TID;<br>1 - blok danych zawiera EPC plus TID (FastID włączony).<br>Bit6:<br>0 - wyłączona faza i Freq;<br>1 - włączona faza i Freq.<br>bit 5 ~ bit 0:<br>długość danych N |         |       |        |                     |

**UWAGA:** Funkcja FastID działa tylko w przypadku wkładek Impinj obsługujących FastID.

### Odczyt danych

Zadaniem tej komendy jest odczyt danych z pamięci zarezerwowanej, pamięci EPC, pamięci TID, pamięci użytkownika tagu. Operacja odczytu rozpoczyna się od zdefiniowanego adresu. Jednostką danych jest słowo.

### Ramka komendy:

| Len  | Adr  | Cmd  | Data[] | CRC-16 |     |
|------|------|------|--------|--------|-----|
| 0xXX | 0xXX | 0x02 | —      | LSB    | MSB |

### Parametry Data[]:

| Data[] |          |      |         |      |         |         |         |         |          |
|--------|----------|------|---------|------|---------|---------|---------|---------|----------|
| ENum   | EPC      | Mem  | WordPtr | Num  | Pwd     | MaskMem | MaskAdr | MaskLen | MaskData |
| 0xXX   | variable | 0xXX | 0xXX    | 0xXX | 4 bytes | 0xXX    | 2 bytes | 0xXX    | variable |

### Definicja parametrów

#### ENum:

0x00 ~ 0x0f – ENum reprezentuje długość numeru EPC. Jednostką ENum jest słowo, a długość EPC powinna być mniejsza niż 15 słów. W tej sytuacji parametry MaskMen, MaskAdr, MaskLen, MaskData powinny pozostać puste.

0xff – zamiast parametrów EPC należy podać parametry MaskMem, MaskAdr, MaskLen, MaskData. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w polu zostanie podana inna wartość.

**EPC:** Numer EPC tagu docelowego. Długość EPC zależy od podanego numeru EPC, a wartość długości musi być liczbą całkowitą.

Jednostką numeru EPC jest słowo. Podczas komunikacji, najbardziej znaczące słowo i najbardziej znaczący bajt w słowie zostaną przesłane jako pierwsze. Niezbędne jest, aby podać FULL EPC w tym poleceniu.

**Mem:** 1 bajt, typ pamięci docelowej.

0x00 – pamięć zarezerwowana

0x01 – pamięć EPC;

0x02 – pamięć TID;

0x03 – pamięć użytkownika.

Wszystkie pozostałe wartości są zarezerwowane. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie podana inna wartość.

**WordPtr:** 1 bajt, początkowy adres słowa celu. Na przykład, 0x00 oznacza, że operacja odczytu rozpoczyna się od 1. słowa (pierwszy 16-bitowy obszar pamięci), 0x01 oznacza, że operacja rozpoczyna się od 2. Słowa itd.

**Num:** 1 bajt, ilość słów, które zostaną odczytane w tej operacji.

Dlatego 0x00 nie jest prawidłową wartością dla tego parametru.

Ponadto wartość Num nie powinna przekraczać 120, tj. maksymalna ilość odczytu wynosi 120 słów. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie podana wartość 0 lub większa niż 120.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to wysokie bajty hasła dostępu, a ostatnie dwa bajty to niskie bajty. Najbardziej znaczący bit 32-bitowego hasła dostępu do tagu jest najbardziej znaczącym bitem w 1. bajcie Pwd (od lewej do prawej).

Podobnie, najmniej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najmniej znaczący bit w czwartym (ostatnim) bajcie Pwd. Hasło dostępu będzie używane do weryfikacji podczas operacji odczytu pamięci zarezerwowanej, gdy powiązana pamięć docelowa jest chroniona hasłem, a hasło dostępu do tagu jest niezerowe. W przeciwnym razie Psw powinno być ustawione na zero lub na prawidłowe hasło dostępu.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć TID;

0x03 – pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli inna wartość zostanie podana w tym polu.

**MaskAdr:** 2 bajty, bitowy adres wejściowy maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące powinny być wypełnione 0.

**UWAGA:**



Jeśli nie podano MaskMem, MaskAdr, MaskLen i MaskData, pełny numer EPC zostanie użyty jako wzorzec maski w wykazie.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd | Status | Data[]           | CRC-16 |      |
|------|------|-------|--------|------------------|--------|------|
| 0xXX | 0xXX | 0x02  | 0x00   | Word1, Word2,... | LSB    | 0xXX |

### Definicja parametrów

Word1, Word2...: jednostką danych w tym polu jest słowo. Każde słowo zawiera 2 bajty, najbardziej znaczący bajt jako pierwszy. Word1 jest pierwszym słowem od adresu początkowego, a drugim słowem jest Word 2 itd.

### Zapis danych

Funkcja tego polecenia polega na zapisywaniu wielu słów do pamięci zarezerwowanej, pamięci TID lub pamięci użytkownika w jednym cyklu komunikacyjnym.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Data[] | CRC-16 |     |
|------|------|------|--------|--------|-----|
| 0xXX | 0xXX | 0x03 | —      | LSB    | MSB |

### Parametry Data[]:

| Data[] |         |          |         |          |          |
|--------|---------|----------|---------|----------|----------|
| WNum   | ENum    | EPC      | Mem     | WordPtr  | Wdt      |
| 0xXX   | 0xXX    | variable | 0xXX    | 0xXX     | variable |
| Pwd    | MaskMem | MaskAdr  | MaskLen | MaskData |          |

|         |      |         |      |          |
|---------|------|---------|------|----------|
| 4 bytes | 0xXX | 2 bytes | 0xXX | variable |
|---------|------|---------|------|----------|

## Definicja parametrów

**WNum:** ilość słów do zapisania. Każde słowo zawiera 2 bajty. Wartość WNum musi być większa niż 0 i mniejsza lub równa 32, a także identyczna z rzeczywistą ilością słowa, które ma zostać zapisane. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu podana zostanie wartość 0 lub nieprawidłowa wartość parametru WNum.

### **ENum:**

0x00 ~ 0x0f – ENum reprezentuje długość numeru EPC. Jednostką ENum jest słowo, a długość EPC powinna być mniejsza niż 15 słów. W tej sytuacji parametry MaskMen, MaskAdr, MaskLen, MaskData powinny pozostać puste.

0xff – zamiast parametrów EPC należy podać parametry MaskMem, MaskAdr, MaskLen, MaskData. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w polu zostanie podana inna wartość.

**EPC:** numer EPC tagu docelowego. Długość EPC zależy od podanego numeru EPC. Jednostką numeru EPC jest słowo. Podczas komunikacji najbardziej znaczące słowo i najbardziej znaczący bajt w słowie zostaną przesłane jako pierwsze. W tym poleceniu należy koniecznie podać PEŁNY numer EPC.

**Mem:** 1 bajt, typ pamięci docelowej.

0x00 – pamięć zarezerwowana

0x01 – pamięć EPC;

0x02 – pamięć TID;

0x03 – pamięć użytkownika.

Wszystkie pozostałe wartości są zarezerwowane. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie podana inna wartość.

**WordPtr:** 1 bajt, początkowy adres słowa celu.

**Wdt:** słowa do zapisania w pamięci tagu. Ilość słów w Wdt musi być identyczna z wartością WNum. Najbardziej znaczący bajt jest pierwszy w słowie. Słowa będą zapisywane do tagu od niższego adresu do wyższego adresu zgodnie z kolejnością w Data[]. Na przykład, WordPtr ma wartość 0x02, pierwsze słowo

w Data[] (od lewej do prawej) zostanie zapisane pod adresem 0x02 w pamięci określonej przez Mem, drugie słowo zostanie zapisane pod adresem 0x03 itd.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty są wysokimi bajtami hasła dostępu, a ostatnie dwa bajty są niskimi bajtami. Najbardziej znaczący bit 32-bitowego hasła dostępu do tagu jest najbardziej znaczącym bitem w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najmniej znaczący bit w

czwartym (ostatnim) bajcie Pwd. Hasło dostępu musi być prawidłowo podane i zostanie zweryfikowane przed operacjami zapisu. Jeśli pamięć docelowa nie jest chroniona hasłem, Pwd może być ustawione na zero.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – Pamięć EPC;

0x02 – pamięć TID;

0x03 – pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli inna wartość zostanie podana w tym polu.

**MaskAdr:** 2 bajty, bitowy adres wejściowy maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące powinny być wypełnione 0.

#### UWAGA:

Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną podane, pełny numer EPC zostanie użyty jako wzorzec maski w inwentaryzacji.

#### Ramka odpowiedzi:

| Len  | Adr  | reCmd | Status | Data[] | CRC-16 |     |
|------|------|-------|--------|--------|--------|-----|
| 0x05 | 0xXX | 0x03  | 0x00   | —      | LSB    | MSB |

#### Wprowadzanie numeru EPC

Funkcją tego polecenia jest zapisanie numeru EPC do tagu. Podczas tej operacji zapisu EPC, tylko jeden tag może być umieszczony w efektywnym obszarze anteny.

| Len  | Adr  | Cmd  | Data[] |         |          | CRC-16 |     |
|------|------|------|--------|---------|----------|--------|-----|
|      |      |      | ENum   | Pwd     | WEPC     |        |     |
| 0xXX | 0xXX | 0x04 | 0xXX   | 4 bytes | variable | LSB    | MSB |

#### Definicja parametrów

**ENum:** 1 bajt, długość numeru EPC. Jednostką ENum jest słowo, a długość EPC może wynosić 0, ale powinna być mniejsza niż 15 słów.

Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie podana inna wartość.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to wysokie bajty hasła dostępu, a ostatnie dwa bajty to niskie bajty. Najbardziej znaczący bit 32-bitowego hasła dostępu do tagu jest najbardziej znaczącym bitem w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najmniej znaczący bit w czwartym (ostatnim) bajcie Pwd. Hasło dostępu będzie używane do weryfikacji podczas operacji odczytu pamięci zarezerwowanej, gdy powiązana pamięć docelowa jest chroniona hasłem, a hasło dostępu do tagu jest niezerowe. W przeciwnym razie Psw powinno być ustawione na zero lub na prawidłowe hasło dostępu.

**WEPC:** numer EPC, który ma zostać zapisany w tagu. Ilość słów w WEPC musi być identyczna z wartością ENum. Minimalna i maksymalna długość WEPC to 1 i 31. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli dane o niewłaściwej długości zostaną dostarczone w tym polu.

| Len  | Adr  | reCmd | Status | Data[] | CRC-16 |     |
|------|------|-------|--------|--------|--------|-----|
| 0x05 | 0xXX | 0x04  | 0x00   | —      | LSB    | MSB |

Ustawienie ochrony przed odczytem/zapisem dla określonej pamięci

Funkcją tego polecenia jest ustawienie stanu ochrony przed odczytem/zapisem dla następujących pamięci

- Pamięć zarezerwowana  
Odczytywalna/zapisywalna bez ochrony, trwale odczytywalna/zapisywalna, odczytywalna/zapisywalna z chroniona hasłem lub trwale nieczytelna/niezapisywalna;
- Pamięć EPC/pamięć zarezerwowana dla użytkownika

Zapisywalna bez ochrony, trwale zapisywalna, zapisywalna z ochroną hasłem lub trwale niezapisywalna;

Pamięć EPC, pamięć zarezerwowana dla użytkownika i pamięć TID są trwale odczytywalne. Ponadto pamięć TID jest tylko do odczytu.

Gdy stan ochrony pamięci zarezerwowanej zostanie ustawiony na trwale czytelny/zapisywalny lub trwale nieczytelny/niezapisywalny, konfiguracja stanu nie będzie w stanie zaakceptować żadnych przyszłych modyfikacji. Podobnie, jeśli status ochrony pamięci EPC, pamięci zarezerwowanej dla użytkownika lub pamięci TID jest ustawiony na trwale zapisywalny / niezapisywalny, konfiguracja statusu nie będzie w stanie zaakceptować żadnych przyszłych modyfikacji. Tag zwróci status błędu w ramce odpowiedzi, jeśli użytkownik spróbuje wykonać przyszłą modyfikację. Aby ustawić określoną pamięć jako chronioną hasłem do odczytu/zapisu lub zmienić status z chronionego hasłem na brak ochrony, należy podać hasło dostępu. W związku z tym przed wykonaniem tej operacji należy upewnić się, że hasło dostępu jest już skonfigurowane dla tagu RFID.

| Len  | Adr  | Cmd  | Data[] | CRC-16 |     |
|------|------|------|--------|--------|-----|
| 0xXX | 0xXX | 0x06 | —      | LSB    | MSB |

| Data[] |          |        |            |         |         |         |         |          |
|--------|----------|--------|------------|---------|---------|---------|---------|----------|
| ENum   | EPC      | Select | SetProtect | Pwd     | MaskMem | MaskAdr | MaskLen | MaskData |
| 0xXX   | variable | 0xXX   | 0xXX       | 4 bytes | 0xXX    | 2 bytes | 0xXX    | variable |

## Definicja parametrów

### ENum:

0x00 ~ 0x0f – ENum oznacza długość liczby EPC. Jednostką ENum jest słowo, a długość EPC powinna być mniejsza niż 15 słów. W

takiej sytuacji parametry MaskMen, MaskAdr, MaskLen, MaskData powinny pozostać puste.

0xff – Zamiast parametrów EPC należy podać MaskMem, MaskAdr, MaskLen, MaskData. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**EPC:** numer EPC znacznika docelowego. Długość świadectwa charakterystyki energetycznej zależy od podanego numeru EPC. Jednostką numeru EPC jest słowo. Podczas komunikacji najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie. Niezbędne jest podanie PEŁNEGO EPC w tym poleceniu.

**Wybierz:** 1 bajt.

0x00 – modyfikacja ochrony przed odczytem/zapisem hasła kill;

0x01 – modyfikacja zabezpieczenia przed odczytem/zapisem hasła dostępu;

0x02 – modyfikacja zabezpieczenia pamięci EPC przed

odczytem/zapisem; 0x03 –

modyfikacja zabezpieczenia pamięci

TID przed odczytem/zapisem; 0x04 –

modyfikacja ochrony pamięci

użytkownika przed

odczytem/zapisem.

Wszystkie inne wartości są zarezerwowane, reader nie wykona polecenia i zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**SetProtect:** wartość SetProtect zależy od wartości Select.

- W przypadku Select = 0x00 lub 0x01, tj. ustawienia kill password lub access password protection wartość SetProtect może być następująca:

0x00 – ustawione na czytelne/zapisywalne bez zabezpieczenia;

0x01 – ustawione na stale czytelne/zapisywalne;

0x02 – ustaw na czytelny/zapisywalny z

ochroną hasłem; 0x03 – ustawione na

trwale nieczytelne/niezapisywalne.

- Dla Select = 0x02, 0x03, 0x04, tj. ustawienia EPC, TID i ochrony pamięci użytkownika, wartość SetProtect może być następująca:

0x00 – ustawione na zapisywalne bez zabezpieczenia;

0x01 – ustawione na zapisywalne na stałe;

0x02 – ustaw na zapisywalny z

ochroną hasłem; 0x03 – ustaw na

stałe niezapisywalne.

Wszystkie inne wartości Select i SetProtect są zarezerwowane, czytnik nie wykona polecenia i zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to górna liczba bajtów hasła dostępu, a ostatnie dwa bajty to dolne bajty. Najbardziej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najbardziej znaczący bit w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczący bit 32-bitowego hasła dostępu tagu jest najmniej znaczącym bitem w 4. (ostatnim) bajcie Pwd. Aby rozpocząć operację, należy podać poprawne hasło dostępu.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć TID;



0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskAdr:** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

#### UWAGA:

Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną dostarczone, pełny numer EPC zostanie użyty jako wzór maski w ekwipunku.

#### Ramka odpowiedzi:

| Len  | Adr  | reCmd | Stan | Dane[] | Przeказnik CRC-16 |     |
|------|------|-------|------|--------|-------------------|-----|
| 0x05 | 0xXX | 0x06  | 0x00 | --     | LSB               | MSB |

#### Wymazywanie bloków

Funkcją tego polecenia jest usunięcie wielu słów z pamięci zarezerwowanej, EPC, TID lub pamięci użytkownika.

#### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przeказnik CRC-16 |     |
|------|------|------|--------|-------------------|-----|
| 0xXX | 0xXX | 0x07 | --     | LSB               | MSB |

#### Parametry danych[]:

| Dane[] |  |      |                 |      |         |         |         |         |
|--------|--|------|-----------------|------|---------|---------|---------|---------|
| EEnum  | Świadectwo charakterystyki energetycznej | Mem  | WordPtr (Słowo) | Num  | Pwd     | MaskMem | MaskAdr | MaskLen |
| 0xXX   | zmienna                                  | 0xXX | 0xXX            | 0xXX | 4 bajty | 0xXX    | 2 bajty | 0       |

#### EEnum:

0x00 ~ 0x0f – EEnum oznacza długość liczby EPC. Jednostką EEnum jest słowo, a długość EPC powinna być mniejsza niż 15 słów. W takiej sytuacji parametry MaskMem, MaskAdr, MaskLen, MaskData powinny pozostać puste.

0xff – Zamiast parametrów EPC należy podać MaskMem, MaskAdr, MaskLen, MaskData. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**EPC:** numer EPC znacznika docelowego. Długość świadectwa charakterystyki energetycznej zależy od podanego numeru EPC. Jednostką numeru EPC jest słowo. Podczas komunikacji najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie. Niezbędne jest podanie PEŁNEGO EPC w tym poleceniu.

**Mem:** 1 bajt, typ pamięci docelowej.

0x00 – pamięć zarezerwowana

0x01 – pamięć EPC;

0x02 – pamięć

TID; 0x03 –

Pamięć

użytkownika.

Wszystkie inne wartości są zarezerwowane. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**WordPtr:** 1 bajt, początkowy adres słowa docelowego. Na przykład 0x01 reprezentuje operację kasowania rozpoczynającą się od 2. słowa (drugi 16-bitowy obszar pamięci) itp. W przypadku operacji wymazywania pamięci EPC minimalna wartość WordPtr powinna wynosić 0x01. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona niewłaściwa wartość.

**Liczba:** 1 bajt, ilość słów do usunięcia w tej operacji. Wymazywanie rozpocznie się od adresów podanych w WordPtr i usunie ilość słów podanych w Num. 0x00 nie jest prawidłową wartością dla tego parametru. Czytnik zwróci stan błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona wartość 0 lub większa niż 120.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to górna liczba bajtów hasła dostępu, a ostatnie dwa bajty to dolne bajty. Najbardziej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najbardziej znaczący bit w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczący bit 32-bitowego hasła dostępu tagu jest najmniej znaczącym bitem w 4. (ostatnim) bajcie Pwd. Hasło dostępu będzie używane do weryfikacji tylko wtedy, gdy powiązana pamięć jest chroniona hasłem. Psw powinno być ustawione na all-zero lub ustawione na poprawne hasło dostępu.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć TID;



0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskAdr:** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

#### UWAGA:

Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną dostarczone, pełny numer EPC zostanie użyty jako wzór maski w ekwipunku.

#### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przeказник<br>CRC-16 |     |
|------|------|----------------------|------|--------|----------------------|-----|
| 0x05 | 0xXX | 0x07                 | 0x00 | --     | LSB                  | MSB |

Odczyt konfiguracji ochrony (zgodnie z numerem EPC)

Funkcją tego polecenia jest ustawienie ochrony przed odczytem dla tagu o określonym numerze EPC, stąd tag będzie nieczytelny dla urządzenia. Urządzenie nie będzie w stanie zapytać o numer EPC tego tagu za pomocą operacji inwentaryzacji. To polecenie jest prawidłowe tylko dla tagów NXP UCODE EPC G2X.

#### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0xXX | 0xXX | 0x08 | --     | LSB                   | MSB |

Parametry danych[]:

| Dane[]   |  |                |             |             |             |               |
|----------|--|----------------|-------------|-------------|-------------|---------------|
| ENu<br>m | Świadectwo<br>charakterysty<br>ki<br>energetyczne<br>j | Pw<br>d        | MaskMe<br>m | MaskAd<br>r | MaskLe<br>n | Dane<br>maski |
| 0xXX     | zmienna  | 4<br>bajt<br>y | 0xXX        | 2 bajty     | 0xXX        | zmienn<br>a   |

**ENum:**

0x00 ~ 0x0f – ENum oznacza długość liczby EPC. Jednostką ENum jest słowo, a długość EPC powinna być mniejsza niż 15 słów. W takiej sytuacji parametry MaskMem, MaskAdr, MaskLen, MaskData powinny pozostać puste.

0xff – Zamiast parametrów EPC należy podać MaskMem, MaskAdr, MaskLen, MaskData.

Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**EPC:** numer EPC znacznika docelowego. Długość świadectwa charakterystyki energetycznej zależy od podanego numeru EPC. Jednostką numeru EPC jest słowo. Podczas komunikacji najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie. Niezbędne jest podanie PEŁNEGO EPC w tym poleceniu.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to górna liczba bajtów hasła dostępu, a ostatnie dwa bajty to dolne bajty. Najbardziej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najbardziej znaczący bit w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczący bit 32-bitowego hasła dostępu tagu jest najmniej znaczącym bitem w 4. (ostatnim) bajcie Pwd. Nie jest możliwe ustawienie stanu ochrony przed odczytem, jeśli tag ma hasło dostępu z zerem, stąd czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostaną dostarczone zera lub niepoprawne hasło dostępu.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć TID;

0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskAdr:** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa  $\text{MaskLen}/8$ . Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

#### **UWAGA:**

Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną dostarczone, pełny numer EPC zostanie użyty jako wzór maski w ekwipunku.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x08                 | 0x00 | --     | LSB                   | MSB |

### Odczyt konfiguracji ochrony (bez numeru EPC)

Funkcją tego polecenia jest skonfigurowanie ochrony przed odczytem dla tagów w efektywnym polu. W przeciwieństwie do poprzedniego polecenia z rozdziału 8.2.8, to polecenie wykona operację na wielu zapytanych znacznikach bez identyfikacji tagu. Aby wykonać operację na wielu tagach, ważne jest, aby hasło dostępu było spójne z tymi tagami. To polecenie jest prawidłowe tylko dla tagów NXP UCODE EPC G2X.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[]  | Przełącznik |     |
|------|------|------|---------|-------------|-----|
|      |      |      | Pwd     | CRC-16      |     |
| 0x08 | 0xXX | 0x09 | 4 bajty | LSB         | MSB |

### Definicja parametru

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to górna liczba bajtów hasła dostępu, a ostatnie dwa bajty to dolne bajty. Najbardziej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najbardziej znaczący bit w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczący bit 32-bitowego hasła dostępu tagu jest najmniej znaczącym bitem w 4. (ostatnim) bajcie Pwd. Nie jest możliwe ustawienie stanu ochrony przed odczytem, jeśli tag ma hasło dostępu z zerem, stąd czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostaną dostarczone zera lub niepoprawne hasło dostępu.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x09                 | 0x00 | --     | LSB                   | MSB |

### Odblokuj ochronę przed odczytem

Funkcją tego polecenia jest odblokowanie ochrony przed odczytem tagu. Tylko jeden tag może być umieszczony w obszarze działania anteny. To polecenie jest prawidłowe tylko dla **tagów** NXP UCODE EPC G2X.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[]  | Przełącznik |     |
|------|------|------|---------|-------------|-----|
|      |      |      | Pwd     | CRC-16      |     |
| 0x08 | 0xXX | 0x0a | 4 bajty | LSB         | MSB |

### Definicja parametru

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to górna liczba bajtów hasła dostępu, a ostatnie dwa bajty to dolne bajty. Najbardziej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najbardziej znaczący bit w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczący bit 32-bitowego hasła dostępu tagu jest najmniej znaczącym bitem w 4. (ostatnim) bajcie Pwd. Istotne jest, aby w tym poleceniu podać poprawne hasło dostępu, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczone nieprawidłowe hasło dostępu.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x0a                 | 0x00 | --     | LSB                   | MSB |

### UWAGA:

W przypadku tagów bez funkcji ochrony przed odczytem domyślny stan zostanie odblokowany.

#### Odczytywanie sprawdzania stanu ochrony

To polecenie NIE JEST W STANIE zidentyfikować, czy określony tag obsługuje funkcję ochrony przed odczytem, polecenie TYLKO o stan ochrony przed odczytem tagu. W przypadku tagów bez funkcji ochrony przed odczytem domyślny stan zostanie odblokowany. Tylko jeden tag może być umieszczony w obszarze działania anteny. To polecenie jest prawidłowe tylko dla tagów NXP UCODE EPC G2X.

#### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0x04 | 0xXX | 0x0b | --     | LSB                   | MSB |

#### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[]             | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------------------|-----------------------|-----|
| 0x06 | 0xXX | 0x0b                 | 0x00 | Czytnik<br>ReadPro | LSB                   | MSB |

#### Definicja parametru

**ReadPro:** 0x00 – ochrona przed odczytem

jest wyłączona dla tagu; 0x01 –  
ochrona przed odczytem jest  
włączona dla tagu.

W przypadku tagów bez funkcji ochrony przed odczytem domyślny stan zostanie odblokowany.

## Konfiguracja EAS

Funkcją tego polecenia jest modyfikowanie lub resetowanie stanu EAS. To polecenie jest prawidłowe tylko dla tagów NXP UCODE EPC G2.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0xXX | 0xXX | 0x0c | --     | LSB                   | MSB |

### Parametry danych[]:

| Dane[]   |  |                |          |             |             |             |               |
|----------|--|----------------|----------|-------------|-------------|-------------|---------------|
| ENum     | Świadectwo<br>charakterys-<br>tyki<br>energetyczn-<br>ej | Pw<br>d        | EA<br>S  | MaskMe<br>m | MaskA<br>dr | MaskL<br>en | Dane<br>maski |
| 0xX<br>X | zmienna  | 4<br>bajt<br>y | 0xX<br>X | 0xXX        | 2 bajty     | 0xXX        | zmien-<br>na  |

### ENum:

0x00 ~ 0x0f – ENum oznacza długość liczby EPC. Jednostką ENum jest słowo, a długość EPC powinna być mniejsza niż 15 słów. W takiej sytuacji parametry MaskMem, MaskAdr, MaskLen, MaskData powinny pozostać puste.

0xff – Zamiast parametrów EPC należy podać MaskMem, MaskAdr, MaskLen, MaskData. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**EPC:** numer EPC znacznika docelowego. Długość świadectwa charakterystyki energetycznej zależy od podanego numeru EPC. Jednostką numeru EPC jest słowo. Podczas komunikacji najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie. Niezbędne jest podanie PEŁNEGO EPC w tym poleceniu.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to górna liczba bajtów hasła dostępu, a ostatnie dwa bajty to dolne bajty. Najbardziej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najbardziej znaczący bit w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczący bit 32-bitowego hasła dostępu tagu jest najmniej znaczącym bitem w 4. (ostatnim) bajcie Pwd. Nie jest możliwe ustawienie EAS, jeśli tag ma zerowe hasło dostępu, stąd czytnik zwróci parametr error status w ramce odpowiedzi, jeśli w tym polu zostaną dostarczone zera lub nieprawidłowe hasło dostępu.

**EAS:** 1 bajt.

bit0 = 0, alert EAS jest  
wyłączony; bit0 = 1,  
alert EAS jest włączony.

bit1 ~ bit7 są zarezerwowane, z 0 jako wartością domyślną.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć TID;

0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskAdr:** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

### NUTA

Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną dostarczone, pełny numer EPC zostanie użyty jako wzór maski w ekwipunku.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0xXX | 0xXX | 0x0c                 | 0x00 | --     | LSB                   | MSB |

### Wykrywanie alertów EAS

Funkcją tego polecenia jest wykrywanie alertu EAS, ważnego tylko dla tagów NXP UCODE EPC G2.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0x04 | 0xXX | 0x0d | --     | LSB                   | MSB |

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x0d                 | 0x00 | --     | LSB                   | MSB |

Zwraca "brak działającego tagu", jeśli alert EAS nie zostanie wykryty.

Zasoby reklamowe z pojedynczym tagiem

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0x04 | 0xXX | 0x0f | --     | LSB                   | MSB |

### Ramka odpowiedzi:

| Len      | Ad<br>r  | reCmd<br>powiedzi<br>ał: | Sta<br>n | Dane[<br>] |         |                            | Przełączni<br>k CRC-16 |         |
|----------|----------|--------------------------|----------|------------|---------|----------------------------|------------------------|---------|
|          |          |                          |          | Mrówk<br>a | Nu<br>m | Identyfikat<br>or EPC      |                        |         |
| 0xX<br>X | 0xX<br>X | 0x0f                     | 0x0<br>1 | 0xXX       | 0x01    | EPC-1<br>(wersja<br>EPC-1) | LS<br>B                | MS<br>B |

Definicja

parametrów

**Mrówka:** wskazuje, która antena zapytała o tag. Na przykład 0x04 to 0000 0100 w systemie binarnym, oznacza to, że antena 3 zapytała o ten konkretny tag. W przypadku 0x08 jest to 0000 1000 w systemie binarnym, co oznacza, że Antena 4 zapytała o informacje o tagu.

**Liczba:** 0x01 (stała)

**Identyfikator EPC:** zapytany tag Dane EPC. Na przykład EPC-1 to długość EPC + numer EPC + **wartość RSSI** pierwszego tagu itp. Wartości długości RSSI i EPC są parametrami jednobajtowymi. Jednostką danych EPC C1 G2 jest słowo. Podczas komunikacji

najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie.

Blokuj polecenie pisania

Funkcją tego polecenia jest zapisywanie wielu słów w pamięci zarezerwowanej, EPC, TID i pamięci użytkownika.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0xXX | 0xXX | 0x10 | --     | LSB                   | MSB |

**Parametry danych[]:**

| Dane[]     |         |  |         |                    |         |
|------------|---------|--|---------|--------------------|---------|
| WNum       | ENum    | Świadectwo<br>charakterystyki<br>energetycznej | Mem     | WordPtr<br>(Słowo) | Wdt     |
| 0xXX       | 0xXX    | zmienna  | 0xXX    | 0xXX               | zmienna |
| Pwd        | MaskMem | MaskAdr  | MaskLen | Dane<br>maski      |         |
| 4<br>bajty | 0xXX    | 2 bajty  | 0xXX    | zmienna            |         |

Definicja

parametrów

**WNum:** ilość słów do napisania. Każde słowo zawiera 2 bajty.

Wartość WNum musi być większa niż 0, a także identyczna z rzeczywistą liczbą słów do napisania. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczone 0 lub nieprawidłowa wartość WNum.

**ENum:**

0x00 ~ 0x0f – ENum oznacza długość liczby EPC. Jednostką ENum jest słowo, a długość EPC powinna być mniejsza niż 15 słów. W takiej sytuacji parametry MaskMem, MaskAdr, MaskLen, MaskData powinny pozostać puste.

0xff – Zamiast parametrów EPC należy podać MaskMem, MaskAdr, MaskLen, MaskData. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**EPC:** numer EPC znacznika docelowego. Długość świadectwa charakterystyki energetycznej zależy od podanego numeru EPC. Jednostką numeru EPC jest słowo. Podczas komunikacji najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie. Niezbędne jest podanie PEŁNEGO EPC w tym poleceniu.

**Mem:** 1 bajt, typ pamięci docelowej.

0x00 – pamięć zarezerwowana

0x01 – pamięć EPC;

0x02 – pamięć

TID; 0x03 –

Pamięć

użytkownika.

Wszystkie inne wartości są zarezerwowane. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**WordPtr:** 1 bajt, początkowy adres słowa docelowego.

**Wdt:** słowa, które mają być zapisane w pamięci tagów. Ilość słów w Wdt musi być identyczna z wartością WNum. Najbardziej znaczący bajt jest pierwszy w słowie. Słowa będą zapisywane do tagowania od niższego adresu do wyższego adresu zgodnie z kolejnością w Data[[]].

Na przykład WordPtr to 0x02, pierwsze słowo w Data[[]] (od lewej do

prawej) zostanie zapisane na adres 0x02 w pamięci określonej przez Mem, drugie słowo zostanie następnie zapisane do 0x03 itd.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to górna liczba bajtów hasła dostępu, a ostatnie dwa bajty to dolne bajty. Najbardziej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najbardziej znaczący bit w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najmniej znaczący bit w 4. (ostatnim) bajcie Pwd. Hasło dostępu musi być poprawnie podane i zostanie zweryfikowane przed operacją zapisu. Jeśli pamięć docelowa nie jest chroniona hasłem, Pwd można ustawić na wartość all-zero.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć TID;

0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskAdr:** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

**UWAGA:**

Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną dostarczone, pełny numer EPC zostanie użyty jako wzór maski w ekwipunku.

#### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x10                 | 0x00 | --     | LSB                   | MSB |

#### Uzyskanie parametrów pracy Monza4QT

To polecenie służy do uzyskania aktualnych parametrów roboczych tagu, ważnego tylko dla **tagów** Monza 4QT firmy Impinj.

#### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0xXX | 0xXX | 0x11 | --     | LSB                   | MSB |

#### Parametry danych[]:

| ENum | Świadectwo<br>charakterystyki<br>energetyczne | Pwd     | MaskMem | MaskAdr | MaskLen | Dane maski |
|------|---|---------|---------|---------|---------|------------|
| 0xXX | zmienna                                       | 4 bajty | 0xXX    | 2 bajty | 0xXX    | zmienna    |

#### ENum:

0x00 ~ 0x0f – ENum oznacza długość liczby EPC. Jednostką ENum jest słowo, a długość EPC powinna być mniejsza niż 15 słów. W takiej sytuacji parametry MaskMem, MaskAdr, MaskLen, MaskData powinny pozostać puste.

0xff – Zamiast parametrów EPC należy podać MaskMem, MaskAdr, MaskLen, MaskData. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**EPC:** numer EPC znacznika docelowego. Długość świadectwa charakterystyki energetycznej zależy od podanego numeru EPC. Jednostką numeru EPC jest słowo. Podczas komunikacji najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie. Niezbędne jest podanie PEŁNEGO EPC w tym poleceniu.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to górna liczba bajtów hasła dostępu, a ostatnie dwa bajty to dolne bajty. Najbardziej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najbardziej znaczący bit w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczący bit 32-bitowego hasła dostępu tagu jest najmniej znaczącym bitem w 4. (ostatnim) bajcie Pwd. Hasło dostępu musi być poprawnie podane w poleceniu.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć TID;

0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskAdr:** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

**UWAGA:**

Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną dostarczone, pełny numer EPC zostanie użyty jako wzór maski w ekwipunku.

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Data[] |                            | Przeказnik<br>CRC-16 |     |
|------|------|----------------------|------|--------|----------------------------|----------------------|-----|
| 0x07 | 0xXX | 0x11                 | 0x00 | NC     | Kontrola<br>odstępu<br>QT. | LSB                  | MSB |
|      |      |                      |      | 0x00   | 1 bajt                     |                      |     |

Definicja parametru

**QTcontrol:** Parametry pracy tagu.

bit0: Bieżące ustawienie strony lustrzanej. 0 – prywatny; 1 – publiczne.

Bit1: Ustawienie ochrony odległości. 0 –

wyłączone; 1 – włączone.

Wszystkie inne wartości są zarezerwowane.

Zmodyfikuj parametry pracy Monza4QT

Funkcją tego polecenia jest modyfikacja bieżących parametrów roboczych tagu, dotyczy to tylko tagów Monza 4QT firmy Impinj.



### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0xXX | 0xXX | 0x12 | --     | LSB                   | MSB |

### Parametry danych[]:

| ENum | Świadectwo<br>charakterystyki<br>energetycznej | Kontrola<br>odstępu<br>QT1 | Kontrola<br>QT0 | Pwd        | MaskMem | MaskAdr | Mask |
|------|--|----------------------------|-----------------|------------|---------|---------|------|
| 0xXX | zmienna  | 0x00                       | 0xXX            | 4<br>bajty | 0xXX    | 2 bajty | 0xXX |

#### ENum:

0x00 ~ 0x0f – ENum oznacza długość liczby EPC. Jednostką ENum jest słowo, a długość EPC powinna być mniejsza niż 15 słów. W takiej sytuacji parametry MaskMem, MaskAdr, MaskLen, MaskData powinny pozostać puste.

0xff – Zamiast parametrów EPC należy podać MaskMem, MaskAdr, MaskLen, MaskData. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**EPC:** numer EPC znacznika docelowego. Długość świadectwa charakterystyki energetycznej zależy od podanego numeru EPC.

Jednostką numeru EPC jest słowo. Podczas komunikacji najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie. Niezbędne jest podanie PEŁNEGO EPC w tym poleceniu.

**QTcontrol:** 0x00 (Constnat), zarezerwowany.

**QTcontrol0:** parametry pracy tagu.

**bit0:** Bieżące ustawienie strony lustrzanej. 0 – prywatny; 1 – publiczne.

**Bit1:** Ustawienie ochrony odległości. 0 – wyłączone; 1 – włączone.

Wszystkie inne wartości są zarezerwowane.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to górna liczba bajtów hasła dostępu, a ostatnie dwa bajty to dolne bajty. Najbardziej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najbardziej znaczący bit w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczący bit 32-bitowego hasła dostępu tagu jest najmniej znaczącym bitem w 4. (ostatnim) bajcie Pwd. Hasło dostępu musi być poprawnie podane w poleceniu.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć TID;

0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskAdr:** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość

MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

**UWAGA:**

Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną dostarczone, pełny numer EPC zostanie użyty jako wzór maski w ekwipunku.

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|----------------------|------|--------|----------------------|-----|
| 0x05 | 0xXX | 0x12                 | 0x00 | --     | LSB                  | MSB |

Rozszerzony odczyt danych z przypisaną maską

Funkcją tego polecenia jest odczyt danych z pamięci zarezerwowanej, pamięci EPC, pamięci TID, pamięci użytkownika tagu. Operacja odczytu rozpoczyna się od zdefiniowanego adresu, a jednostką danych jest słowo.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|------|--------|----------------------|-----|
| 0xXX | 0xXX | 0x15 | --     | LSB                  | MSB |

**Parametry danych[]:**

| Dane[] |  |      |                    |      |            |         |         |    |
|--------|--|------|--------------------|------|------------|---------|---------|----|
| ENum   | Świadectwo<br>charakterystyki<br>energetycznej | Mem  | WordPtr<br>(Słowo) | Num  | Pwd        | MaskMem | MaskAdr | Ma |
| 0xXX   | zmienna  | 0xXX | 2 bajty            | 0xXX | 4<br>bajty | 0xXX    | 2 bajty | 0  |

ENum:

0x00 ~ 0x0f – ENum oznacza długość liczby EPC. Jednostką ENum jest słowo, a długość EPC powinna być mniejsza niż 15 słów. W takiej sytuacji parametry MaskMen, MaskAdr, MaskLen, MaskData powinny pozostać puste.

0xff – Zamiast parametrów EPC należy podać MaskMem, MaskAdr, MaskLen, MaskData. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**EPC:** numer EPC znacznika docelowego. Długość świadectwa charakterystyki energetycznej zależy od podanego numeru EPC. Jednostką numeru EPC jest słowo. Podczas komunikacji najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie. Niezbędne jest podanie PEŁNEGO EPC w tym poleceniu.

**Mem:** 1 bajt, typ pamięci docelowej.

0x00 – pamięć zarezerwowana

0x01 – pamięć EPC;

0x02 – pamięć

TID; 0x03 –

Pamięć

użytkownika.

Wszystkie inne wartości są zarezerwowane. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**WordPtr:** 2 bajty, początkowy adres słowa docelowego. Na przykład 0x0000 reprezentują, że operacja odczytu rozpoczyna się od pierwszego słowa (pierwszego 16-bitowego obszaru pamięci), 0x0001 reprezentuje operację rozpoczynającą się od 2. słowa itd.

**Liczba:** 1 bajt, ilość słów do odczytania w tej operacji, dlatego 0x00 nie jest prawidłową wartością dla tego parametru. Ponadto wartość Num nie powinna przekraczać 120, czyli maksymalna ilość czytanych słów to 120. Czytnik zwróci stan błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona wartość 0 lub większa niż 120.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to wysokie bajty hasła dostępu, a ostatnie dwa bajty to dolne bajty. Najbardziej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najbardziej znaczący bit w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczący bit 32-bitowego hasła dostępu do tagu jest najmniej znaczącym bitem w 4. (ostatnim) bajcie Pwd. Hasło dostępu będzie używane do weryfikacji tylko podczas operacji odczytu pamięci zarezerwowanej, gdy powiązana pamięć docelowa jest chroniona hasłem, a hasło dostępu do tagu jest niezerowe. W przeciwnym razie Psw powinno być ustawione na all-zero lub ustawione na poprawne hasło dostępu.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć TID;

0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskAdr:** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

**UWAGA:**

Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną dostarczone, pełny numer EPC zostanie użyty jako wzór maski w ekwipunku.

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[]            | Przeказnik<br>CRC-16 |     |
|------|------|----------------------|------|-------------------|----------------------|-----|
| 0xXX | 0xXX | 0x15                 | 0x00 | Słowo1,Słowo2,... | LSB                  | MSB |

Definicja parametru

**Słowo1, Słowo2...:** Jednostką tego pola jest słowo. Każde słowo zawiera 2 bajty, najpierw najbardziej znaczący bajt. Słowo1 to 1. słowo od adresu początkowego, a 2. słowo to Słowo 2 itd.

Rozszerzony odczyt danych z przypisaną maską

Funkcją tego polecenia jest zapisanie wielu słów do pamięci zarezerwowanej, pamięci TID lub pamięci użytkownika w jednym cyklu komunikacyjnym.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|------|--------|----------------------|-----|
| 0xXX | 0xXX | 0x16 | --     | LSB                  | MSB |

**Parametry danych[]:**

|        |
|--------|
| Dane[] |
|--------|

| WNum    | ENum    | Świadectwo charakterystyki energetycznej | Mem     | WordPtr (Słowo) | Wdt     |
|---------|---------|--|---------|-----------------|---------|
| 0xXX    | 0xXX    | zmienna                                  | 0xXX    | 2 bajty         | zmienna |
| Pwd     | MaskMem | MaskAdr                                  | MaskLen | Dane maski      |         |
| 4 bajty | 0xXX    | 2 bajty                                  | 0xXX    | zmienna         |         |

**WNum:** ilość słów do napisania. Każde słowo zawiera 2 bajty.

Wartość WNum musi być większa niż 0 i mniejsza lub równa 32, a także identyczna z rzeczywistą liczbą słów do napisania. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczone 0 lub nieprawidłowa wartość WNum.

**ENum:**

0x00 ~ 0x0f – ENum oznacza długość liczby EPC. Jednostką ENum jest słowo, a długość EPC powinna być mniejsza niż 15 słów. W takiej sytuacji parametry MaskMen, MaskAdr, MaskLen, MaskData powinny pozostać puste.

0xff – Zamiast parametrów EPC należy podać MaskMem, MaskAdr, MaskLen, MaskData. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**EPC:** numer EPC znacznika docelowego. Długość świadectwa charakterystyki energetycznej zależy od podanego numeru EPC. Jednostką numeru EPC jest słowo. Podczas komunikacji najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie. Niezbędne jest podanie PEŁNEGO EPC w tym poleceniu.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć TID;

0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**WordPtr:** 2 bajty, najpierw najbardziej znaczący bajt, początkowy adres słowa docelowego.

**Wdt:** słowa, które mają być zapisane w pamięci tagów. Ilość słów w Wdt musi być identyczna z wartością WNum. Najbardziej znaczący bajt jest pierwszy w słowie. Słowa będą zapisywane do tagowania od niższego adresu do wyższego adresu zgodnie z kolejnością w Data[]. Na przykład WordPtr to 0x02, pierwsze słowo w Data[] (od lewej do prawej) zostanie zapisane na adres 0x02 w pamięci określonej przez Mem, drugie słowo zostanie następnie zapisane do 0x03 itd.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej pierwsze dwa bajty to górna wartość hasła dostępu, a ostatnie dwa bajty to dolna wartość bajtu. Najbardziej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najbardziej znaczący bit w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najmniej znaczący bit w 4. (ostatnim) bajcie Pwd. Hasło dostępu musi być poprawnie podane i zostanie zweryfikowane przed operacją zapisu. Jeśli pamięć docelowa nie jest chroniona hasłem, Pwd można ustawić na wartość all-zero.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć TID;



0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskAdr:** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

#### UWAGA:

Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną dostarczone, pełny numer EPC zostanie użyty jako wzór maski w ekwipunku.

#### Ramka poleceń:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|----------------------|------|--------|----------------------|-----|
| 0x05 | 0xXX | 0x16                 | 0x00 | --     | LSB                  | MSB |

#### Inwentaryzacja z buforem pamięci

Polecenie inwentaryzacji służy do wyszukiwania/identyfikacji tagów RFID i przechowuje zapytane dane tagu w buforze pamięci czytnika.

Główna różnica między inwentaryzacją z buforem pamięci a bez niego (patrz rozdział 8.2.1) polega na tym, że dzięki zastosowaniu

wewnętrznego bufora pamięci, czytnik może stale pytać o tagi w czasie inwentaryzacji. Na koniec każdego czasu inwentaryzacji czytnik poinformuje o całkowitej liczbie tagów przechowywanych w buforze pamięci (wielokrotne użycie tagu również zwiększy ilość), a także o ilość tagów zapytanych w bieżącym inwentarzu. Użytkownik może uzyskać dostęp do danych z bufora pamięci za pomocą poleceń pobierania danych znaczników, uzyskiwania ilości znaczników lub czyszczenia bufora pamięci.

Domyślny czas inwentaryzacji to 2 sekundy, można go zmodyfikować za pomocą niestandardowego polecenia czytnika "Modyfikuj czas inwentaryzacji czytnika" (patrz rozdział 8.4.4) lub przeprowadzić tymczasową konfigurację za pomocą opcjonalnego parametru w inwentarzu. Prawidłowy zakres czasu inwentaryzacji wynosi od  $3 * 100$  ms do  $255 * 100$  ms (rzeczywisty czas odpowiedzi może być nieco większy niż wartość ustawienia, a różnica ta będzie się wahać od 0 do 75 ms).

#### UWAGA

- Ustawienie nierozsądnego krótkiego czasu inwentaryzacji może spowodować awarię spisu tagów.
- Bufor zostanie wyczyszczony, jeśli z hosta zostanie dostarczone polecenie "inwentaryzacja bez bufora pamięci".

#### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|------|--------|----------------------|-----|
| 0xXX | 0xXX | 0x18 | --     | LSB                  | MSB |

#### Parametry danych[]:

|        |
|--------|
| Dane[] |
|--------|

| QValue  | Sesja  | MaskMem         | MaskAdr | MaskLen | Dane maski | AdrTID powiedz iał: | LenTID |
|---------|--------|-----------------|---------|---------|------------|---------------------|--------|
| 0xXX    | 0xXX   | 0xXX            | 2 bajty | 0xXX    | zmien na   | 0xXX                | 0xXX   |
| Dane[ ] |        |                 |         |         |            |                     |        |
| Cel     | Mrówka | Czas skanowania |         |         |            |                     |        |
| 0xXX    | 0xXX   | 0xXX            |         |         |            |                     |        |

**QValue:** 1 bajt.

bit7: zarezerwowane, ustawione na 0.

bit6: Wskaźnik

strategii. 0 –  
strategia  
ogólna; 1 –  
Strategia  
specjalna.

bit5 ~ bit0: pierwotna wartość Q inwentarza tagów EPC, ustawienie wartości Q powinno być zgodne z regułą:  $2^Q \approx$  całkowitą liczbę tagów w efektywnym polu. Zakres wartości Q wynosi 0 ~ 15, jeśli w tym polu zostanie dostarczona inna wartość, czytnik zwróci stan błędu parametru w ramce odpowiedzi.

**Sesja:** 1 bajt, wartość sesji inwentarza tagów EPC.

0x00 – zastosuj S0 jako wartość sesji;  
0x01 – zastosuj S1 jako wartość sesji;  
0x02 – zastosuj S2 jako wartość sesji;

---

0x03 – zastosuj S3 jako wartość sesji;  
0xff – zastosuj inteligentną konfigurację czytnika (dotyczy tylko w inwentarzu EPC).

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;  
0x02 – pamięć TID;  
0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskAdr:** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

**AdrTID:** adres wejściowy pamięci TID inwentarza. Jeśli polecenie dostarczyło ten parametr i LenTID w tej samej ramce, tryb inwentaryzacji powinien mieć wartość TID, a nie EPC.

**LenTID:** długość danych dla operacji inwentaryzacji TID, prawidłowy zakres LenTID to 0 ~ 15.

Wszystkie inne wartości są rezerwowane, a czytnik zwróci stan błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie

---

dostarczona inna wartość. Jeśli polecenie dostarczyło ten parametr i AdrTID, tryb inwentaryzacji powinien mieć wartość TID, a nie EPC.

**Cel (parametr opcjonalny):** 1 bajt, wartość docelowa inwentarza tagów EPC. 0x00 – zastosuj A jako wartość docelową; 0x01 – zastosuj B jako wartość docelową.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**Ant (parametr opcjonalny):** 1 bajt, wybór anteny dla aktualnego inwentarza.

0x80 – antena 1;  
0x81 – antena 2;  
0x82 – antena  
3; 0x83 –  
antena 4.

**ScanTime (parametr opcjonalny):** 1 bajt, konfiguracja czasu inwentaryzacji dla bieżącej inwentaryzacji.

Czytnik ustawi maksymalny czas pracy jako ScanTime\*100ms.

**NUTA:**

1. Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną dostarczone, żaden wzór maski nie będzie uwzględniony w inwentarzu
2. Target, Ant i ScanTime są parametrami opcjonalnymi, wszystkie trzy parametry muszą być dostarczone w jednej ramce poleceń. Czytnik zastosuje ustawienie domyślne dla tych parametrów, dopóki wszystkie trzy ustawienia parametrów nie zostaną prawidłowo

dostarczone z hosta, nowe ustawienia zaczną obowiązywać dla śledzenia spisu. **Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[]     |               | Przeказnik<br>CRC-16 |     |
|------|------|----------------------|------|------------|---------------|----------------------|-----|
|      |      |                      |      | Licznik    | Numer<br>tagu |                      |     |
| 0x09 | 0xXX | 0x18                 | 0x00 | 2<br>bajty | 2<br>bajty    | LSB                  | MSB |

**BufferCount:** 2 bajty, całkowita ilość tagów przechowywanych w buforze pamięci, tagi z identycznymi danymi EPC/TID będą traktowane jako jeden tag. BufferCount to suma ilości tagów z wielu inwentaryzacji, dopóki bufor pamięci nie zostanie wyczerpany.

**TagNum:** 2 bajty, ilość tagu zapytanego w bieżącym spisie, tag używany wiele razy również zwiększy ilość.

### Mieszanie zasobów reklamowych

Polecenie inwentaryzacji służy do wyszukiwania tagów RFID z odpowiednim protokołem w efektywnym polu i uzyskiwania określonych informacji o tagu po zapytaniu o numer EPC tagu.

W zależności od okoliczności użytkownik może modyfikować ustawienie maksymalnego czasu pracy (czas inwentaryzacji). Czytnik zwróci wynik inwentaryzacji w predefiniowanym czasie inwentaryzacji. Jeśli czytnik nie zakończy procesów inwentaryzacji dla wszystkich znaczników w efektywnym polu w predefiniowanym czasie inwentaryzacji, czytnik nie przeprowadzi dalszej inwentaryzacji po przekroczeniu czasu inwentaryzacji. Czytnik zgłosi wszystkie już poszukiwane tagi, a także bieżący (nieukończony) stan zapasów do hosta. Reader będzie wtedy czekał na następne polecenie od hosta.

Domyślny czas inwentaryzacji to 2 s, można go zmodyfikować za pomocą niestandardowego polecenia czytelnika "modyfikuj czas inwentaryzacji czytelnika" (patrz rozdział 8.4.4). Prawidłowy zakres czasu inwentaryzacji wynosi od 3 \* 100 ms do 255 \* 100 ms (rzeczywisty czas odpowiedzi może być nieco większy niż wartość ustawienia, a różnica ta będzie się wahać od 0 do 75 ms).

**UWAGA:**

Ustawienie nierozsądnego krótkiego czasu inwentaryzacji może spowodować awarię spisu tagów.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[] | Przeказnik CRC-16 |     |
|------|------|------|--------|-------------------|-----|
| 0xXX | 0xXX | 0x19 | --     | LSB               | MSB |

**Parametry danych[]:**

| Dane[] |       |         |         |         |            |
|--------|-------|---------|---------|---------|------------|
| QValue | Sesja | MaskMem | MaskAdr | MaskLen | Dane maski |
| 0xXX   | 0xXX  | 0xXX    | 2 bajty | 0xXX    | zmienna    |

  

| Dane[]     |            |                      |         |      |        |                 |
|------------|------------|----------------------|---------|------|--------|-----------------|
| Odczyt Mem | Odczyt Adr | ReadLen (Odczyt Len) | Pwd     | Cel  | Mrówka | Czas skanowania |
| 0xXX       | 2 bajty    | 0xXX                 | 4 bajty | 0xXX | 0xXX   | 0xXX            |

QValue: 1 bajt.

---

bit7: Flaga pakietu danych statystycznych.

0 – Po inwentaryzacji NIE dostarczaj pakietu danych statystycznych z procesu inwentaryzacji;

1 – Po inwentaryzacji dostarcz pakiet danych statystycznych procesu inwentaryzacji Bit4: flaga danych fazy.

0 –wyłączać;

1 – włącz bit3 ~ bit0: pierwotna wartość Q inwentarza tagów EPC, ustawienie wartości Q powinno być zgodne z regułą:  $2^Q \approx$  całkowitą liczbę tagów w efektywnym polu. Zakres wartości Q wynosi 0 ~ 15, jeśli w tym polu zostanie dostarczona inna wartość, czytnik zwróci stan błędu parametru w ramce odpowiedzi.

**Sesja:** 1 bajt, wartość sesji inwentarza tagów EPC.

0x00 – zastosuj S0 jako wartość sesji;

0x01 – zastosuj S1 jako wartość sesji;

0x02 – zastosuj S2 jako wartość sesji;

0x03 – zastosuj S3 jako wartość sesji;

0xff – zastosuj inteligentną konfigurację czytnika (dotyczy tylko w inwentarzu EPC).

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskMem (parametr opcjonalny):** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć TID;

0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

---

**MaskAdr (parametr opcjonalny):** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen (parametr opcjonalny):** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData (parametr opcjonalny):** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ .

Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

**ReadAdr:** 2 bajty, adres wejściowy celu, najpierw najbardziej znaczący bajt. Na przykład 0x0000 reprezentują, że operacja odczytu rozpoczyna się od pierwszego słowa (pierwszego 16-bitowego obszaru pamięci), 0x0001 reprezentuje operację rozpoczynającą się od 2. słowa itd.

**ReadLen:** 1 bajt, ilość słów do odczytania w tej operacji, minimalna długość to 1 słowo, a maksymalna to 120 słów. 0x00 nie jest prawidłową wartością dla tego parametru; ponadto wartość Num nie powinna przekraczać 120. Czytnik zwróci stan błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona wartość 0 lub większa niż 120.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to górna liczba bajtów hasła dostępu, a ostatnie dwa bajty to dolne bajty. Najbardziej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najbardziej znaczący bit w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczący bit 32-bitowego hasła dostępu do tagu jest najmniej znaczącym bitem w 4. (ostatnim) bajcie Pwd. Hasło dostępu będzie używane do weryfikacji tylko podczas operacji odczytu pamięci zarezerwowanej, gdy powiązana pamięć docelowa jest chroniona hasłem, a hasło dostępu do tagu jest niezerowe. W przeciwnym razie Psw powinno

---

być ustawione na all zero lub ustawione na poprawne hasło dostępu.

**Cel (parametr opcjonalny):** 1 bajt, wartość docelowa inwentarza tagów EPC. 0x00 – zastosuj A jako wartość docelową; 0x01 – zastosuj B jako wartość docelową.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**Ant (parametr opcjonalny):** 1 bajt, wybór anteny dla aktualnego inwentarza.

0x80 – antena 1;  
0x81 – antena 2;  
0x82 – antena  
3; 0x83 –  
antena 4.

**ScanTime (parametr opcjonalny):** 1 bajt, konfiguracja czasu inwentaryzacji dla bieżącej inwentaryzacji.

Czytnik ustawi maksymalny czas pracy jako  $\text{ScanTime} * 100\text{ms}$ .

**NUTA:**

1. Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną dostarczone, żaden wzór maski nie będzie uwzględniony w inwentarzu
2. Target, Ant i ScanTime są parametrami opcjonalnymi, wszystkie trzy parametry muszą być dostarczone w jednej ramce poleceń. Czytnik zastosuje ustawienie domyślne dla tych parametrów, dopóki wszystkie trzy ustawienia parametrów nie zostaną prawidłowo

dostarczone z hosta, nowe ustawienia zaczną obowiązywać dla śledzenia spisu.

(1) Status odpowiedzi to 0x26, odpowiedź powinna być następująca:

| Len      | Adr      | reCmd<br>powiedzia<br>ł: | Sta<br>n | Dane[]     |                         |                      | Przeakaźni<br>k CRC-16 |     |
|----------|----------|--------------------------|----------|------------|-------------------------|----------------------|------------------------|-----|
|          |          |                          |          | Mrówk<br>a | Szybkoś<br>ć<br>odczytu | Łączn<br>a<br>liczba |                        |     |
| 0xX<br>X | 0xX<br>X | 0x19                     | 0xX<br>X | 0xXX       | 2 bajty                 | 4<br>bajty           | LSB                    | MSB |

**Status:** status odpowiedzi, szczegółowy opis jest pokazany w poniższej tabeli

| Stan | Opis  |
|------|---|
| 0x26 | Po inwentaryzacji dostarcz dane statystyczne dotyczące bieżącego procesu inwentaryzacji |

**Ant:** wskazuje, która antena zapytała o tag. Na przykład 0x04 to 0000 0100 w systemie binarnym, oznacza to, że antena 3 zapytała o ten konkretny tag. W przypadku 0x08 jest to 0000 1000 w systemie binarnym, co oznacza, że Antena 4 zapytała o informacje o tagu.

**ReadRate:** wskaźnik identyfikacji tagu dla bieżącej inwentaryzacji (czas udanej identyfikacji / s), znacznik identyfikowany wiele razy również wydłuży czas udanej identyfikacji.

**TotalCount:** łączna ilość tagów wykrytych w bieżącym spisie, tag, do którego użytkownik uzyskuje wielokrotny dostęp, również zwiększy łączną kwotę.

(2) Status odpowiedzi nie jest 0x26, odpowiedź powinna być następująca:

| Len      | Adr      | reCmd<br>powiedział<br>: | Sta<br>n | Dane[<br>] |         |   | Przekaźni<br>k CRC-16 |     |
|----------|----------|--------------------------|----------|------------|---------|---|-----------------------|-----|
|          |          |                          |          | Mrówk<br>a | Nu<br>m | Pakiet<br>danych                          |                       |     |
| 0xX<br>X | 0xX<br>X | 0x19                     | 0xX<br>X | 0xXX       | 0xXX    | Pakiet-1,<br>Pakiet-2,<br>Pakiet-<br>3... | LSB                   | MSB |

**Status:** status odpowiedzi, szczegółowy opis jest pokazany w poniższej tabeli

| Stan | Opis  |
|------|---|
| 0x01 | Operacja zakończona, czytnik zgłosi wszystkie żądane informacje o tagach do hosta   |
| 0x02 | Przekroczono limit czasu zapasów, operacja została przerwana. Czytnik zgłosi wszystkie już zapytane tagi do hosta   |
| 0x03 | Czytnik nie jest w stanie odpowiedzieć na wszystkie dane w jednej ramce, dalsze dane będą przesyłane w kolejnych ramkach.   |
| 0x04 | Czytnik ukończył tylko części inwentarza, ale zabrakło mu miejsca w pamięci ze względu na ilość tagów. Czytnik zgłosi wszystkie już zapytane tagi, a także tę wartość stanu do hosta. |
| 0xF8 | Wykryto błąd anteny, bieżąca antena może być odłączona.   |

**Mrówka:** wskazuje, która antena zapytała o tag. Na przykład 0x04 to 0000 0100 w systemie binarnym, oznacza to, że antena 3 zapytała o ten konkretny tag. W przypadku 0x08 jest to 0000 1000 w systemie binarnym, co oznacza, że Antena 4 zapytała o informacje o tagu.

---

**Liczba:** ilość pakietów danych informacji o tagach zawartych w tej odpowiedzi.

**Pakiet danych:** poszukiwane dane tagu, Pakiet-1 to dane pierwszego pakietu, Pakiet-2 to dane drugiego pakietu itp. Format typowego pakietu jest pokazany poniżej:

| Paczka ět        |      |         |                      |        |         |
|------------------|------|---------|----------------------|--------|---------|
| Parametr pakietu | Len  | Dane    | Wskaźnik RSSI (RSSI) | faza   | Freq    |
| 0xXX             | 0xXX | zmienna | 0xXX                 | 4 byts | 3 bajty |

Parametry pakietu danych:

**PacketParam:** 1 bajt, parametry tego pakietu. bit7: wskaźnik typu tego pakietu danych.

- 0 – ten pakiet danych zawiera numer EPC tagu;
- 1 – ten pakiet danych zawiera poszukiwane dane tagu (numer EPC tego tagu jest dostarczany w ostatnim pakiecie danych).

bit6 ~ bit0: numer seryjny pakietu danych. Zakres numeru seryjnego wynosi 0 ~ 127. Na początku inwentaryzacji każdego miksu numer seryjny pakietu zostanie zrestartowany od 0 i zwiększy się dla każdego przesłanego pakietu danych. Gdy numer seryjny osiągnie maksymalną wartość 127, zostanie ponownie przeliczony od 0.

**Len:** 1 bajt, długość danych w tym pakiecie. jeśli bit6 =1, włącz fazę i częstotliwość, te dane mają 7 bajtów,

**Dane:** numer EPC lub poszukiwane dane z określonego obszaru tagu. Jednostką danych tagu **EPC C1 G2** jest słowo. Podczas komunikacji

---

najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie.

**RSSI:** 1 bajt, siła sygnału czytnika w momencie zapytania o tag.

**Faza:** 4 bajty, jeśli **bit Lena 6 = 1**, 2 bajty rozpoczynają fazę, 2 bajty kończą fazę;

**Freq:** 3 bajty, jeśli **bit Lena 6=1**, jednostką jest khz;

**UWAGA:**

Pakiet danych zostanie przesłany zgodnie z zamówieniem podanym przez EPC. Dla każdego tagu czytnik najpierw prześle tag EPC w początkowym pakiecie danych, dalsze dane tagu zostaną przesłane w drugim pakiecie danych. Te dwa pakiety będą miały sekwencyjny numer seryjny.

W przypadku, gdy czytnik nie odczyta danych tagu po przesłaniu pakietu EPC, czytnik nie wczyta żadnego pakietu danych tagu. Czytnik będzie nadal pytać o następny tag i ponownie uruchomić normalny proces inwentaryzacji mieszanki.

Inwentaryzacja z numerem EPC

Funkcją tego polecenia jest zapytanie tagu o określony EPC. Czytnik odpowie tylko na dane z tagu z odpowiednim EPC. Wszystkie inne tagi z innym EPC zostaną zignorowane.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|------|--------|----------------------|-----|
| 0xXX | 0xXX | 0x1a | --     | LSB                  | MSB |

## Parametry danych[]:

| Dane[]          |                        |                          |                                   |
|-----------------|------------------------|--------------------------|-----------------------------------|
| Typ dopasowania | MatchLen (Dopasowanie) | Przesunięcie dopasowania | EPCData (Konferencja Klimatyczna) |
| 0xXX            | 2 bajty                | 2 bajty                  | zmienna                           |

MatchType: typ dopasowania numeru EPC.

0 – zapytania o etykietę z odpowiednim EPC;

1 – zapytaj o wszystkie inne tagi z różnymi EPC.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MatchLen:** 2 bajty, najpierw najbardziej znaczący bajt, długość bitu zdefiniowanego numeru EPC. Zakres MatchLen wynosi 1 ~ 196 bitów.

**MatchOffset:** 2 bajty, najpierw najbardziej znaczący bajt. Adres wejściowy zdefiniowanego EPC (jednostka: bity).

Zakres MatchOffset wynosi 0 ~ 495 bitów.

### UWAGA:

Suma **MatchLen** i **MatchOffset** nie powinna przekraczać 496 bitów. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona nieprawidłowa wartość.

**EPCData:** zdefiniowane dane o numerze EPC. Długość EPCData jest równa  $\text{MatchLen}/8$ . Jeśli MatchLen nie jest wielokrotnością liczby całkowitej 8, długość EPCData jest równa  $\text{int}[\text{MatchLen}/8]+1$ .

Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

### Ramka odpowiedzi:

|    |    |  |     |      |  |
|----|----|--|-----|------|--|
| Le | Ad |  | Sta | Dane |  |
| n  | r  |  | n   | []   |  |

|          |          | reCmd<br>powiedzi<br>ał: |          | Mrów<br>ka | Nu<br>m  | Identyfika<br>tor EPC      | Przekaż<br>nik<br>CRC-16 |         |
|----------|----------|--------------------------|----------|------------|----------|----------------------------|--------------------------|---------|
| 0xX<br>X | 0xX<br>X | 0x1a                     | 0xX<br>X | 0xXX       | 0xX<br>X | EPC-1<br>(wersja<br>EPC-1) | LS<br>B                  | MS<br>B |

**Status:** status odpowiedzi, szczegółowy opis jest pokazany w poniższej tabeli

| Stan | Opis  |
|------|---|
| 0x01 | Operacja zakończona, czytnik zgłosi wszystkie żądane informacje o tagach do hosta   |
| 0x02 | Przekroczono limit czasu zapasów, operacja została przerwana. Czytnik zgłosi wszystkie już zapytane tagi do hosta         |
| 0x03 | Czytnik nie jest w stanie odpowiedzieć na wszystkie dane w jednej ramce, dalsze dane będą przesyłane w kolejnych ramkach. |
| 0xF8 | Wykryto błąd anteny, bieżąca antena może być odłączona.   |

**Mrówka:** wskazuje, która antena zapytała o tag. Na przykład 0x04 to 0000 0100 w systemie binarnym, oznacza to, że antena 3 zapytała o ten konkretny tag. W przypadku 0x08 jest to 0000 1000 w systemie binarnym, co oznacza, że Antena 4 zapytała o informacje o tagu.

**Num:** ilość tagu EPC zawartego w tej odpowiedzi.

**EPC ID:** zapytane dane EPC/TID. Na przykład EPC-1 to długość EPC/TID + numer EPC/dane TID + wartość RSSI pierwszego znacznika itp. Wartości długości RSSI i EPC są parametrami jednobajtowymi. Jednostką danych EPC C1 G2 jest słowo. Podczas komunikacji

---

najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie.

### Zapasy QT

To polecenie służy do sprawdzania prywatnego numeru EPC tagu Monza4QT z publicznym dublowaniem. Polecenie wyszukuje tagi RFID z odpowiednim protokołem. W celu przeprowadzenia dalszej operacji na nowym tagu z nieznanym EPC, użytkownik powinien najpierw uzyskać numer EPC tagu za pomocą komendy inwentaryzacyjnej.

W zależności od okoliczności użytkownik może modyfikować ustawienie maksymalnego czasu pracy (czas inwentaryzacji). Czytnik zwróci wynik inwentaryzacji w predefiniowanym czasie inwentaryzacji. Jeśli czytnik nie zakończy procesów inwentaryzacji dla wszystkich znaczników w efektywnym polu w predefiniowanym czasie inwentaryzacji, czytnik nie przeprowadzi dalszej inwentaryzacji po przekroczeniu czasu inwentaryzacji. Czytnik zgłosi wszystkie już poszukiwane tagi, a także bieżący (nieukończony) stan zapasów do hosta. Reader będzie wtedy czekał na następne polecenie od hosta.

Domyślny czas inwentaryzacji to 1s, można go zmodyfikować za pomocą niestandardowego polecenia czytnika "modyfikuj czas inwentaryzacji czytnika" (patrz rozdział 8.4.4). Prawidłowy zakres czasu inwentaryzacji wynosi od  $3 * 100$  ms do  $255 * 100$  ms (rzeczywisty czas odpowiedzi może być nieco większy niż wartość ustawienia, a różnica ta będzie się wahać od 0 do 75 ms).

**Uwaga:** ustawienie nierozsądnego krótkiego czasu inwentaryzacji może spowodować awarię spisu tagów.

**Ramka poleceń:**

|       |      |      |        |                       |     |
|-------|------|------|--------|-----------------------|-----|
| Len   | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
| 0x0XX | 0xXX | 0x1b | --     | LSB                   | MSB |

### Parametry danych[]:

| Dane[] |       |      |        |                    |
|--------|-------|------|--------|--------------------|
| QValue | Sesja | Cel  | Mrówka | Czas<br>skanowania |
| 0xXX   | 0xXX  | 0xXX | 0xXX   | 0xXX               |

**QValue:** 1 bajt.

bit7: Flaga pakietu danych statystycznych.

0 – Po inwentaryzacji NIE dostarczaj pakietu danych statystycznych z procesu inwentaryzacji;

1 – Po inwentaryzacji dostarcz pakiet danych statystycznych procesu inwentaryzacji bit6 ~ bit0: oryginalna wartość Q inwentarza tagów EPC, ustawienie wartości Q powinno być zgodne z zasadą:  $2^Q \approx$  całkowitą liczbę tagów w efektywnym polu. Zakres wartości Q wynosi 0 ~ 15, jeśli w tym polu zostanie dostarczona inna wartość, czytnik zwróci stan błędu parametru w ramce odpowiedzi.

**Sesja:** 1 bajt, wartość sesji inwentarza tagów EPC.

0x00 – zastosuj S0 jako wartość sesji;

0x01 – zastosuj S1 jako wartość sesji;

0x02 – zastosuj S2 jako wartość sesji;

0x03 – zastosuj S3 jako wartość sesji;

0xff – zastosuj inteligentną konfigurację czytnika (dotyczy tylko w inwentarzu EPC).

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

---

**Cel (parametr opcjonalny):** 1 bajt, wartość docelowa inwentarza tagów EPC. 0x00 – zastosuj A jako wartość docelową; 0x01 – zastosuj B jako wartość docelową.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**Ant (parametr opcjonalny):** 1 bajt, wybór anteny dla aktualnego inwentarza.

0x80 – antena 1;  
0x81 – antena 2;  
0x82 – antena  
3; 0x83 –  
antena 4.

Wszystkie inne wartości są rezerwowane, a czytnik zwróci stan błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**ScanTime (parametr opcjonalny):** 1 bajt, konfiguracja czasu inwentaryzacji dla bieżącej inwentaryzacji.

Czytnik ustawi maksymalny czas pracy jako  $\text{ScanTime} * 100\text{ms}$ .

**UWAGA:**

1. Target, Ant i ScanTime są parametrami opcjonalnymi, wszystkie trzy parametry muszą być dostarczone w jednej ramce poleceń. Czytnik zastosuje ustawienie domyślne dla tych parametrów, dopóki wszystkie trzy ustawienia parametrów nie zostaną prawidłowo dostarczone z hosta, nowe ustawienia zaczną obowiązywać dla śledzenia spisu.

(1) Status odpowiedzi to 0x26, odpowiedź powinna być następująca:

| Len      | Adr      | reCmd<br>powiedzia<br>ł: | Sta<br>n | Dane[]     |                         |                      | Przeakaźni<br>k CRC-16 |     |
|----------|----------|--------------------------|----------|------------|-------------------------|----------------------|------------------------|-----|
|          |          |                          |          | Mrówk<br>a | Szybkoś<br>ć<br>odczytu | Łączn<br>a<br>liczba |                        |     |
| 0xX<br>X | 0xX<br>X | 0x1b                     | 0xX<br>X | 0xXX       | 2 bajty                 | 4<br>bajty           | LSB                    | MSB |

**Status:** status odpowiedzi, szczegółowy opis znajduje się w poniższej tabeli,

| Stan | Opis   |
|------|--|
| 0x26 | Po inwentaryzacji dostarcz dane statystyczne dotyczące ostatniego procesu inwentaryzacji |

**Mrówka:** wskazuje, która antena zapytała o tag. Na przykład 0x04 to 0000 0100 w systemie binarnym, oznacza to, że antena 3 zapytała o ten konkretny tag. W przypadku 0x08 jest to 0000 1000 w systemie binarnym, co oznacza, że Antena 4 zapytała o informacje o tagu.

**ReadRate:** wskaźnik identyfikacji tagu dla bieżącej inwentaryzacji (czas udanej identyfikacji / s), znacznik identyfikowany wiele razy również wydłuży czas udanej identyfikacji.

**TotalCount:** łączna ilość tagów wykrytych w bieżącym spisie, tag, do którego użytkownik uzyskuje wielokrotny dostęp, również zwiększy łączną kwotę.

(2) Status odpowiedzi nie jest 0x26, odpowiedź powinna być następująca:

| Len | Adr | reCmd<br>powiedzia<br>ł: | Sta<br>n | Dane[]     |         |                       | Przeakaźni<br>k CRC-16 |  |
|-----|-----|--------------------------|----------|------------|---------|-----------------------|------------------------|--|
|     |     |                          |          | Mrówk<br>a | Nu<br>m | Identyfikat<br>or EPC |                        |  |

|          |          |      |          |      |      |                            |     |     |
|----------|----------|------|----------|------|------|----------------------------|-----|-----|
| 0xX<br>X | 0xX<br>X | 0x1b | 0xX<br>X | 0xXX | 0xXX | EPC-1, EPC-<br>2, EPC-3... | LSB | MSB |
|----------|----------|------|----------|------|------|----------------------------|-----|-----|

**Status:** status odpowiedzi, szczegółowy opis znajduje się w poniższej tabeli,

| Stan | Opis  |
|------|---|
| 0x01 | Operacja zakończona, czytnik zgłosi wszystkie żądane informacje o tagach do hosta   |
| 0x02 | Przekroczono limit czasu zapasów, operacja została przerwana. Czytnik zgłosi wszystkie już zapytane tagi do hosta         |
| 0x03 | Czytnik nie jest w stanie odpowiedzieć na wszystkie dane w jednej ramce, dalsze dane będą przesyłane w kolejnych ramkach. |
| 0x04 | Czytnik ukończył tylko części inwentarza, ale zabrakło mu miejsca w pamięci   |
|      | ze względu na ilość tagów. Czytnik zgłosi wszystkie już zapytane tagi, a także tę wartość stanu do hosta.                 |

**Mrówka:** wskazuje, która antena zapytała o tag. Na przykład 0x04 to 0000 0100 w systemie binarnym, oznacza to, że antena 3 zapytała o ten konkretny tag. W przypadku 0x08 jest to 0000 1000 w systemie binarnym, co oznacza, że Antena 4 zapytała o informacje o tagu.

**Liczba:** ilość EPC/TID zawarta w tej odpowiedzi.

**EPC ID:** zapytane dane EPC/TID. Na przykład EPC-1 to długość EPC/TID + numer EPC/dane TID + **wartość RSSI** pierwszego znacznika itp.

Wartości długości RSSI i EPC są parametrami jednobajtowymi.

Jednostką danych EPC C1 G2 jest słowo. Podczas komunikacji najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie.

---

## Polecenia 18000-6B

### Zasoby reklamowe z pojedynczym tagiem

Funkcją tego polecenia jest zapytanie o pojedynczy tag. Jeśli w polu efektywnym umieszczono wiele tagów, czytnik może nie zapytać o żaden tag.

#### Ramka poleceń:

| Len  | Adr  | Cmd  | Przełącznik<br>CRC-16 |     |
|------|------|------|-----------------------|-----|
| 0x04 | 0xXX | 0x50 | LSB                   | MSB |

#### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[]              |              | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|---------------------|--------------|-----------------------|-----|
|      |      |                      |      | Mrówka              | ID           | LSB                   | MSB |
| 0x10 | 0xXX | 0x50                 | 0x00 | 0Xxx<br>powiedział: | 10<br>bajtów |                       |     |

**Mrówka:** wskazuje, która antena zapytała o tag. Na przykład 0x04 to 0000 0100 w systemie binarnym, oznacza to, że antena 3 zapytała o ten konkretny tag. W przypadku 0x08 jest to 0000 1000 w systemie binarnym, co oznacza, że Antena 4 zapytała o informacje o tagu.

**ID:** 10 bajtów, 1. bajt to długość UID, 0x08. 2. ~ 9. bajt to numer UID tagu, najpierw najmniej znaczący bajt. 10. bajt to RSSI.

Zapytaj o tagi zgodnie z określonym warunkiem

Funkcją tego polecenia jest wyszukiwanie tagów zgodnie z określonym warunkiem.

#### Ramka poleceń:

| Len | Adr |  | Dane[] |  |
|-----|-----|--|--------|--|
|-----|-----|--|--------|--|

|      |     | Cmd  | Warunek | Adres | Maska | Word_data | Przebieg CRC-16 |     |
|------|-----|------|---------|-------|-------|-----------|-----------------|-----|
| 0x0f | 0xX | 0x51 | 0xXX    | 0xXX  | 0xXX  | 8 bajtów  | LSB             | MSB |

**Warunek:** warunek inwentaryzacji;

0x00 – równy referencji;

0x01 – nie jest równa referencji;

0x02 – większa niż referencyjna

0x03 – mniejsza od referencyjnej

**Adres:** adres porównywanych danych

**Maska:** maska, określ dane do porównania.

**Word\_data:** wartość referencji. Każdy bit w masce odpowiada bajtowi w Word\_data. Najbardziej znaczący bit (bit7) odpowiada pierwszemu bajtowi w Word\_data (od lewej do prawej). W związku z tym najmniej znaczący bit (bit0) odpowiada ostatniemu bajtowi w Word\_data (od lewej do prawej). Word\_data są danymi referencyjnymi do porównania z danymi w tagu zaczyna się od lokalizacji podanej w polu Adres. 1. bajt Word\_data zostanie porównany z magazynem danych w polu Adres, a ostatni bajt Word\_data zostanie porównany z magazynem danych w polu Adres+7. Ustaw odpowiadający bajtowi bit maski na 1, czytnik wykona porównanie na tym konkretnym bajcie. Ustaw bit na 0, żadne porównanie nie zostanie wykonane na tym konkretnym bajcie.

**Ramka odpowiedzi:**

| Len | Adr | reCmd<br>powiedział<br>: | Stan | Mrówka | Num | Dane[ ] | Przebieg<br>CRC-16 |
|-----|-----|--------------------------|------|--------|-----|---------|--------------------|
|     |     |                          |      |        |     |         |                    |

|          |          |      |          |      |      |                  |         |         |
|----------|----------|------|----------|------|------|------------------|---------|---------|
| 0xX<br>X | 0xX<br>X | 0x51 | 0xX<br>X | 0xXX | 0xXX | UID1,<br>UID2... | LS<br>B | MS<br>B |
|----------|----------|------|----------|------|------|------------------|---------|---------|

**Status:** status odpowiedzi, szczegółowy opis jest pokazany w poniższej tabeli

| Stan | Opis  |
|------|---|
| 0x15 | Operacja zakończona, czytnik zgłosi wszystkie zapytane informacje o tagach do hosta.  |
| 0x16 | Przekroczono limit czasu zapasów, operacja została przerwana. Czytnik zgłosi wszystkie już zapytane tagi do hosta.  |
| 0x17 | Czytnik nie jest w stanie odpowiedzieć na wszystkie dane w jednej ramce, dalsze dane będą przesyłane w kolejnych ramkach.   |
| 0x18 | Czytnik ukończył tylko części inwentarza, ale zabrakło mu miejsca w pamięci ze względu na ilość tagów. Czytnik zgłosi wszystkie już zapytane tagi, a także tę wartość stanu do hosta. |
| 0xF8 | Wykryto błąd anteny, bieżąca antena może być odłączona.   |

**Mrówka:** wskazuje, która antena zapytała o tag. Na przykład 0x04 to 0000 0100 w systemie binarnym, oznacza to, że antena 3 zapytała o ten konkretny tag. W przypadku 0x08 jest to 0000 1000 w systemie binarnym, co oznacza, że Antena 4 zapytała o informacje o tagu.

**Num:** ilość UID zawarta w tej odpowiedzi.

**Data[ ]:** UID tagu. Długość każdego bloku danych UID wynosi 10 bajtów. 1. bajt to wartość długości numeru UID, czyli 0x08. 2. ~ 9. bajt to numer UID tagu, najpierw najmniej znaczący bajt. Dziesiąty bajt to wartość RSSI. Dla Num = 0, tj. nie podano żadnych danych UID, Data[ ] jest pusta.

---

## Dane czytnika

Funkcją tego polecenia jest odczyt wielu bajtów z określonego adresu.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] |             |      | Przeказник<br>CRC-16 |     |
|------|------|------|--------|-------------|------|----------------------|-----|
|      |      |      | Adres  | ID          | Num  |                      |     |
| 0x0e | 0xXX | 0x52 | 0xXX   | 8<br>bajtów | 0xXX | LSB                  | MSB |

**Adres:** adres wejściowy celu. Prawidłowy zakres wartości adresu to 0 ~ 223. Czytnik zwróci parametr error status w ramce odpowiedzi, jeśli zostanie dostarczony nieprawidłowy adres.

**Liczba:** ilość słów, które zostaną odczytane w tej operacji. Prawidłowy zakres wartości Num to 1 ~ 32. Jeśli suma wartości Adres i Liczba jest większa niż 224, wartość Num jest równa 0 lub większa niż 32, zwraca stan błędu parametru w ramce odpowiedzi.

**ID:** identyfikator znacznika docelowego, 8 bajtów, najpierw najmniej znaczący bajt.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przeказник<br>CRC-16 |     |
|------|------|----------------------|------|--------|----------------------|-----|
| 0xXX | 0xXX | 0x52                 | 0x00 | Dane   | LSB                  | MSB |

**Dane:** zapytane dane z tagu, najpierw najmniej znaczący bajt.

---

## Zapis danych

Funkcją tego polecenia jest zapisanie wielu bajtów do określonego znacznika.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] |             |         | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-------------|---------|-----------------------|-----|
|      |      |      | Adres  | ID          | Wdata   | LSB                   | MSB |
| 0xXX | 0xXX | 0x53 | 0xXX   | 8<br>bajtów | zmienna | LSB                   | MSB |

**Adres:** 1 bajt, adres bajtu docelowego. Zakres adresów to 8 ~ 223.

Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**ID:** 8 bajtów, numer identyfikacyjny znacznika docelowego, najpierw najmniej znaczący bajt.

**Wdata:** dane, które mają zostać zapisane w tagu. Długość Wdata nie powinna przekraczać 32 bajtów. Najbardziej znaczący bajt zostanie zapisany do najniższego bajtu od wpisu określonego w polu Adres. Jeśli długość Wdata jest równa 0 (brak danych) lub przekracza 32, czytnik zwróci status błędu parametru w ramce odpowiedzi. Poza tym, jeśli suma Address i Wdata przekroczy 224, czytnik również zwróci błąd parametru.

### Ramka poleceń:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x53                 | 0x00 | --     | LSB                   | MSB |

---

## Uzyskiwanie stanu blokady

Funkcją tego polecenia jest uzyskanie stanu blokady określonego bajtu.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] |             | Przełącznik |     |
|------|------|------|--------|-------------|-------------|-----|
|      |      |      | Adres  | ID          | CRC-16      |     |
| 0x0d | 0xXX | 0x54 | 0xXX   | 8<br>bajłów | LSB         | MSB |

**Adres:** 1 bajt, adres bajtu docelowego. Zakres adresów wynosi 0 ~ 223. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**ID:** 8 bajłów, numer identyfikacyjny znacznika docelowego. Bajt najmniej znaczący powinien być przesłany jako pierwszy.

### Ramka poleceń:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[]          | Przełącznik |     |
|------|------|----------------------|------|-----------------|-------------|-----|
|      |      |                      |      |                 | CRC-16      |     |
| 0x06 | 0xXX | 0x54                 | 0x00 | Stan<br>blokady | LSB         | MSB |

**LockState:** 0x00: bajt jest odblokowany; 0x01: bajt jest zablokowany.

## Bajt blokady

Funkcją tego polecenia jest zablokowanie określonego bajtu w tagu.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] |             | Przełącznik<br>CRC-16 |      |
|------|------|------|--------|-------------|-----------------------|------|
|      |      |      | Adres  | ID          |                       |      |
| 0x0d | 0xXX | 0x55 | 0xXX   | 8<br>bajtów | 0x0d                  | 0xXX |

**Adres:** 1 bajt, adres bajtu docelowego. Zakres adresów to 8 ~ 223. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**ID:** 8 bajtów, numer identyfikacyjny znacznika docelowego. Bajt najmniej znaczący powinien być przesłany jako pierwszy.

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x55                 | 0x00 | --     | LSB                   | MSB |

Indywidualne polecenia czytnika

### Uzyskiwanie informacji o czytniku

Funkcją tego polecenia jest uzyskanie informacji o czytniku, w tym adresu czytnika, wersji oprogramowania układowego, kodu modelu czytnika, obsługiwanych protokołów, pasma częstotliwości roboczej, mocy RF, czasu inwentaryzacji, stanu brzęczyka, GPIO do powiadamiania itp.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0x04 | 0xXX | 0x21 | --     | LSB                   | MSB |

**Ramka odpowiedzi:**

| Len | Adr | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |
|-----|-----|----------------------|------|--------|-----------------------|
|     |     |                      |      |        |                       |

|      |      |      |      |  |     |     |
|------|------|------|------|--|-----|-----|
| 0x0f | 0xXX | 0x21 | 0x00 | Wersja, Typ,<br>Tr_Type,<br>dmaxfre,<br>dminfre, Power,<br>Scntm, Ant,<br>Zarezerwowany,<br>Zarezerwowany,<br>CheckAnt | LSB | MSB |
|------|------|------|------|--|-----|-----|

**Parametry danych[]:**

| Parametry              | Długość (bajty) | Opis   |
|------------------------|-----------------|--|
| Wersja                 | 2               | Wersja oprogramowania układowego, wysoki bajt określa główny numer wersji, a dolny bajt określa numer podwersji.               |
| Typ                    | 1               | Kod modelu czytnika, 0x0c to RRU2881M / RRU2881LITE  |
| Tr_Type                | 1               | Obsługiwane prototypy.<br>bit1 = 1: obsługuje 18000-6C. bit0 = 1: obsługuje 18000-6B.<br>Wszystkie inne bity są zarezerwowane. |
| dmaxfre<br>powiedział: | 1               | bit7 ~ bit6: konfiguracja pasma częstotliwości; bit5 ~ bit0: maksymalny punkt częstotliwości.                                  |
| dminfre<br>powiedział: | 1               | bit7 ~ bit6: konfiguracja pasma częstotliwości; bit5 ~   |

|                   |   |   |
|-------------------|---|---|
|                   |   | bit0: minimalny punkt częstotliwości.   |
| Moc               | 1 | Moc wyjściowa RF, zakres od 0 do 30.  |
| SCNTM powiedział: | 1 | Czas inwentaryzacji.<br>Czytnik odpowie na polecenie inwentaryzacji dostarczone z hosta w tym konkretnym czasie inwentaryzacji. |
| Mrówka            | 1 | Konfiguracja anteny.  |
| Zastrzeżony       | 1 | Zastrzeżony.  |
| Zastrzeżony       | 1 | Zastrzeżony.  |
| Szachant          | 1 | Konfiguracja sprawdzania anteny 0: sprawdzanie anteny wyłączone; 1: sprawdzanie anteny włączone.                                |

**Tabela konfiguracji pasma częstotliwości**

| MaxFre(bit 7) | MaxFre(bit 6) | MinFre(bit 7) | MinFre(bit 6) | Pasmo częstotliwości |
|---------------|---------------|---------------|---------------|----------------------|
| 0             | 0             | 0             | 0             | Zastrzeżony          |
| 0             | 0             | 0             | 1             | Chiński zespół       |
| 0             | 0             | 1             | 0             | Pasmo amerykańskie   |
| 0             | 0             | 1             | 1             | Koreański zespół     |

|     |     |     |     |                      |
|-----|-----|-----|-----|----------------------|
| 0   | 1   | 0   | 0   | Pasmo UE             |
| 0   | 1   | 0   | 1   | Zastrzeżony          |
| 0   | 1   | 1   | 0   | Zespół z Ukrainy     |
| 0   | 1   | 1   | 1   | Zespół z Peru        |
| 1   | 0   | 0   | 0   | Chiński zespół       |
| 1   | 0   | 0   | 1   | Pasmo EU3            |
| 1   | 0   | 1   | 0   | Zespół tajwański     |
| 1   | 1   | 0   | 0   | Pasmo amerykański e3 |
| ... | ... | ... | ... | ...                  |
| 1   | 1   | 1   | 1   | Zastrzeżony          |

Zmodyfikuj częstotliwość roboczą

Funkcją tego polecenia jest wybór pasma częstotliwości, modyfikacja maksymalnych i minimalnych punktów częstotliwości. Maksymalny punkt częstotliwości musi być większy lub równy minimalnemu punktowi częstotliwości.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[] |        | Przeказnik CRC-16 |     |
|------|------|------|--------|--------|-------------------|-----|
|      |      |      | MaxFre | MinFre |                   |     |
| 0x06 | 0xXX | 0x22 | 0xXX   | 0xXX   | LSB               | MSB |

**MaxFre:** 1 bajt.

bit7 ~ bit6: konfiguracja pasma częstotliwości; bit5 ~ bit0:

---

maksymalny punkt  
częstotliwości.

**MinFre:** 1 bajt.

bit7 ~ bit6: konfiguracja pasma

częstotliwości; bit5 ~ bit0:

minimalny punkt

częstotliwości.

**UWAGA:**

Maksymalny punkt częstotliwości musi być większy lub równy minimalnemu punktowi częstotliwości. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**Tabela konfiguracji pasma częstotliwości**

| MaxFre(bit 7) | MaxFre(bit 6) | MinFre(bit 7) | MinFre(bit 6) | Pasma częstotliwości |
|---------------|---------------|---------------|---------------|----------------------|
| 0             | 0             | 0             | 0             | Zastrzeżony          |
| 0             | 0             | 0             | 1             | Chiński zespół       |
| 0             | 0             | 1             | 0             | Pasma amerykańskie   |
| 0             | 0             | 1             | 1             | Koreański zespół     |
| 0             | 1             | 0             | 0             | Pasma UE             |
| 0             | 1             | 0             | 1             | Zastrzeżony          |
| 0             | 1             | 1             | 0             | Zespół z Ukrainy     |
| 0             | 1             | 1             | 1             | Zespół z Peru        |

|     |     |     |     |                      |
|-----|-----|-----|-----|----------------------|
| 1   | 0   | 0   | 0   | Chiński zespół       |
| 1   | 0   | 0   | 1   | Pasmo EU3            |
| 1   | 0   | 1   | 0   | Zespół tajwański     |
| 1   | 1   | 0   | 0   | Pasmo amerykańskie 3 |
| ... | ... | ... | ... | ...                  |
| 1   | 1   | 1   | 1   | Zastrzeżony          |

### Ramka odpowiedzi:

| Len  | Adr  | reCmd powiedział: | Stan | Dane[] | Przeказnik CRC-16 |     |
|------|------|-------------------|------|--------|-------------------|-----|
| 0x05 | 0xXX | 0x22              | 0x00 | --     | LSB               | MSB |

### Wzory dla różnych pasm częstotliwości:

Chiński zespół2:  $F_s = 920,125 + N * 0,25$  (MHz), gdzie  $N \in [0, 19]$ .

Pasmo amerykańskie:  $F_s = 902,75 + N * 0,5$  (MHz), gdzie  $N \in [0, 49]$ .

amerykańskie:

Koreański zespół:  $F_s = 917,1 + N * 0,2$  (MHz), gdzie  $N \in [0, 31]$ .

zespół:

Pasmo EU:  $F_s = 865,1 + N * 0,2$  (MHz), gdzie  $N \in [0, 14]$ .

Ukraina zespół:  $F_s = 868,0 + N * 0,1$  (MHz), gdzie  $N \in [0, 6]$ .

Zespół z Peru:  $F_s = 916,2 + N * 0,9$  (MHz), gdzie  $N \in [0, 11]$ .

Chiński zespół1:  $F_s = 840,125 + N * 0,25$  (MHz), gdzie  $N \in [0, 19]$ .

Pasmo EU3:  $F_s = 865,7 + N * 0,6$  (MHz), gdzie  $N \in [0, 3]$ .

Pasmo amerykańskie3:  $F_s = 902 + N * 0,5$  (MHz), gdzie  $N \in [0, 52]$ .

amerykańskie3:

Zespół z Tajwanu:  $F_s = 922,25 + N * 0,5$  (MHz), gdzie  $N \in [0, 11]$ .

Tajwanu:

---

## Zmień adres czytnika

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przełącznik |     |
|------|------|------|--------|-------------|-----|
|      |      |      | adres  | CRC-16      |     |
| 0x05 | 0xXX | 0x24 | 0xXX   | LSB         | MSB |

**Adr:** nowe ustawienie adresu czytnika. 0xFF nie jest prawidłowym adresem, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli 0xFF zostanie dostarczony w tym polu.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x24                 | 0x00 | --     | LSB                   | MSB |

### UWAGA:

Czytnik odpowie na to polecenie adresem ORIGINAL.

## Modyfikowanie czasu inwentaryzacji czytnika

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[]             | Przełącznik<br>CRC-16 |     |
|------|------|------|--------------------|-----------------------|-----|
| 0x05 | 0xXX | 0x25 | Czas<br>skanowania | LSB                   | MSB |
|      |      |      | 0xXX               |                       |     |

**Czas skanowania:** czas inwentaryzacji. Czytnik zmodyfikuje maksymalny czas odpowiedzi zgodnie z wartością zdefiniowaną przez użytkownika (0\*100ms ~ 255\*100ms), a czytnik zastosuje to nowe ustawienie dla przyszłych inwentaryzacji. Domyślne ustawienie czasu skanowania to 0x14 (co odpowiada 20\*100ms). Prawidłowe

ustawienie czasu skanowania to 0x00 ~ 0xff (co odpowiada 3 \* 100 ms ~ 255 \* 100 ms).

**UWAGA:** rzeczywisty czas odpowiedzi może być nieco większy niż wartość ustawienia, a różnica ta będzie się wahać od 0 do 75 ms. W przypadku czasu skanowania = 0 czas operacji nie jest ograniczony do momentu zakończenia inwentaryzacji wszystkich tagów.

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x25                 | 0x00 | --     | LSB                   | MSB |

Modyfikowanie szybkości transmisji szeregowej

Funkcją tego polecenia jest modyfikacja szybkości transmisji dla portu szeregowego.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[]                 | Przełącznik<br>CRC-16 |     |
|------|------|------|------------------------|-----------------------|-----|
| 0x05 | 0xXX | 0x28 | szybkość<br>transmisji | LSB                   | MSB |
|      |      |      | 0xXX                   |                       |     |

Definicja parametrów **baudrate**: nowe ustawienie szybkości transmisji, opcje szybkości transmisji to 0/1/2/5/6 (patrz poniższa tabela, aby uzyskać prawidłową szybkość transmisji), wszystkie inne wartości są zarezerwowane.

*Tabela konfiguracyjna szybkości transmisji*

|                     |                                    |
|---------------------|------------------------------------|
| wartość<br>baudrate | Aktualna<br>szybkość<br>transmisji |
|---------------------|------------------------------------|

|   |                               |
|---|-------------------------------|
| 0 | 9600<br>punktów<br>bazowych   |
| 1 | 19200<br>punktów<br>bazowych  |
| 2 | 38400<br>punktów<br>bazowych  |
| 5 | 57600<br>punktów<br>bazowych  |
| 6 | 115200<br>punktów<br>bazowych |

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x28                 | 0x00 | --     | LSB                   | MSB |

**NUTA:**

Odpowiedź zostanie dostarczona przy użyciu ustawienia szybkości transmisji ORIGINAL. Nowe ustawienie szybkości transmisji zacznie obowiązywać w NASTĘPNYM KOMUNIKACIE.

---

## Zmodyfikuj moc RF

Funkcją tego polecenia jest modyfikacja mocy RF czytnika.

Prawidłowa wartość tego parametru to 0 ~ 30. Przy ustawieniu 30 moc wyjściowa wynosi około 1W.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[]             | Przeказnik<br>CRC-16 |     |
|------|------|------|--------------------|----------------------|-----|
| 0x05 | 0xXX | 0x2F | Pwr<br>powiedział: | LSB                  | MSB |
|      |      |      | 0xXX               |                      |     |

**Pwr:** 1 bajt, parametry mocy RF.

bit0 ~ bit6: ustawienie mocy RF, prawidłowa wartość tego parametru to 0 ~ 30. Przy ustawieniu 30 moc wyjściowa wynosi około 1W.

bit7: stan zachowania konfiguracji po wyłączeniu zasilania. 0 – konfiguracja zachowana po wyłączeniu czytnika; 1 – konfiguracja nie jest zachowywana.

### Ramka poleceń:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|----------------------|------|--------|----------------------|-----|
| 0x05 | 0xXX | 0x2F                 | 0x00 | --     | LSB                  | MSB |

## Sterowanie brzęczykiem/diodą LED

Funkcją tego polecenia jest sterowanie diodą TAG i brzęczykiem (share with GPO1 pin) w celu wykonania określonych czynności.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[]  |            |      | Przełącznik<br>CRC-16 |     |
|------|------|------|---------|------------|------|-----------------------|-----|
|      |      |      | ActiveT | Cichy<br>T | Razy |                       |     |
| 0x07 | 0xXX | 0x33 | 0xXX    | 0xXX       | 0xXX | LSB                   | MSB |

**ActiveT:** Dioda LED TAG i brzęczyk (pin GPO1 niski poziom TTL) czas włączenia to ActiveT \* 50ms. Zakres ActiveT wynosi 0 ~ 255.

Ustawienie domyślne to 0.

**SilentT:** dioda LED TAG i brzęczyk (pin GPO1 niski poziom TTL) czas wyłączenia to SilentT \* 50ms. Zakres SilentT wynosi 0 ~ 255.

Ustawienie domyślne to 0.

**Czasy:** czas działania diody LED TAG i brzęczyka, zakres czasów wynosi 0 ~ 255. Ustawienie domyślne to 0.

#### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x33                 | 0x00 | ---    | LSB                   | MSB |

#### Konfiguracja multipleksowania anten

Funkcją tego polecenia jest modyfikacja konfiguracji anteny czytnika.

#### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
|      |      |      | Mrówka |                       |     |
| 0x05 | 0xXX | 0x3f | 0xXX   | LSB                   | MSB |

**Ant:** informacje o konfiguracji anteny. 1 antena to minimalne wymaganie do działania.

bit0: 1 bit

konfiguracyjny

anteny. 0 – wyłącz  
 antenę 1; 1 – Włącz  
 antenę 1. bit1: bit  
 konfiguracyjny  
 anteny 2. 0 – wyłącz  
 antenę 2; 1 – Włącz  
 antenę 2. bit2: bit  
 konfiguracyjny  
 anteny 3. 0 – wyłącz  
 antenę 3; 1 – Włącz  
 antenę 3. bit3: bit  
 konfiguracyjny  
 anteny 4. 0 – wyłącz antenę 4;  
 1 – Włącz antenę 4.  
 bit4 ~ bit6: Zarezerwowane, wartość domyślna to 0.  
 bit7: stan zachowania konfiguracji.  
 0 – konfiguracja zachowywana po  
 wyłączeniu czytnika; 1 – konfiguracja  
 nie została zachowana.

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|----------------------|------|--------|----------------------|-----|
| 0x05 | 0xXX | 0x3f                 | 0x00 | --     | LSB                  | MSB |

**Włącz/wyłącz brzęczyk**

Funkcją tego polecenia jest włączanie/wyłączanie brzęczyka (udostępnianie z pinem GPO1).

**Ramka poleceń:**

| Len | Adr | Cmd | Dane[] |  |
|-----|-----|-----|--------|--|
|     |     |     |        |  |

|      |      |      | Sygnal<br>dźwiękowy | Przeказnik<br>CRC-16 |     |
|------|------|------|---------------------|----------------------|-----|
| 0x05 | 0xXX | 0x40 | 0Xxx<br>powiedział: | LSB                  | MSB |

**BeepEn:** wybór trybu pracy

brzęczyka. bit0: 0 – wyłącz

brzęczyk;

1 – włącz brzęczyk, brzęczyk będzie emitował sygnał dźwiękowy przy każdej operacji tagu sukcesu (niski impuls wyjściowy pinu GPO1).

bit1 ~ bit7 są zarezerwowane, wartość domyślna to 0.

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|----------------------|------|--------|----------------------|-----|
| 0x05 | 0xXX | 0x40                 | 0x00 | --     | LSB                  | MSB |

Sterowanie GPIO

Funkcją tego polecenia jest sterowanie wyjściem GPIO. Domyślnym wyjściem jest wysoki poziom czasu wygaśnięcia.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[]            | Przeказnik<br>CRC-16 |     |
|------|------|------|-------------------|----------------------|-----|
|      |      |      | Pin-<br>wyjściowy |                      |     |
| 0x05 | 0xXX | 0x46 | 0xXX              | LSB                  | MSB |

Definicja parametru

**OutputPin:** stan wyjścia GPIO (pin Out1 do Out2). Bit0 ~ bit3 kontrolują odpowiednio Out1 do Out3. Bit2 ~ bit7 są zarezerwowane.

**Ramka odpowiedzi:**

| Len | Adr | reCmd<br>powiedział: | Stan | Dane[] | Przeказnik<br>CRC-16 |  |
|-----|-----|----------------------|------|--------|----------------------|--|
|-----|-----|----------------------|------|--------|----------------------|--|

|      |      |      |      |        |     |     |
|------|------|------|------|--------|-----|-----|
| 0x05 | 0xXX | 0x46 | 0x00 | —<br>— | LSB | MSB |
|------|------|------|------|--------|-----|-----|

### Uzyskiwanie stanu GPIO

Funkcją tego polecenia jest uzyskanie stanów wejściowych i wyjściowych ramki poleceń GPIO:

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0x04 | 0xXX | 0x47 | —      | LSB                   | MSB |

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[]            | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|-------------------|-----------------------|-----|
|      |      |                      |      | Pin-<br>wyjściowy |                       |     |
| 0x06 | 0xXX | 0x47                 | 0x00 | 0xXX              | LSB                   | MSB |

### Definicja parametru

**OutputPin:** stany wejścia i wyjścia GPIO. Bit0 reprezentuje stan pinu IN1, bity 4 i bit5 reprezentują odpowiednio stan Out1 do Out2.

Wszystkie inne wartości są zarezerwowane.

### Uzyskaj unikalny numer seryjny czytelnika

Funkcją tego polecenia jest uzyskanie unikalnego numeru seryjnego czytelnika.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0x04 | 0xXX | 0x4c | —      | LSB                   | MSB |

### Ramka odpowiedzi:

| Len | Adr |  | Stan | Dane[] |  |
|-----|-----|--|------|--------|--|
|     |     |  |      |        |  |

|      |      | reCmd<br>powiedział: |      | SeriaNo | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|---------|-----------------------|-----|
| 0x09 | 0xXX | 0x4c                 | 0x00 | 4bajty  | LSB                   | MSB |

**SeriaNo:** 4 bajty, unikalny numer seryjny czytnika.

### Modyfikowanie funkcji niestandardowej tagu

Funkcją tego polecenia jest uruchomienie niestandardowych narzędzi niektórych określonych tagów, aby osiągnąć określoną funkcję tagu.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[]        | Przełącznik<br>CRC-16 |     |
|------|------|------|---------------|-----------------------|-----|
|      |      |      | Typ<br>wkładu |                       |     |
| 0x05 | 0xXX | 0x3a | 0xXX          | LSB                   | MSB |

### Definicja parametru

**InlayType:** typ tagu, prawidłowa wartość tego parametru to 0 ~ 254.

0 – Domyślna wartość tego parametru, typ tagu reprezentuje, nie jest określona.

1 – uruchom funkcję Peek tagu Monza4QT (status tagu tymczasowo zmieni się z publicznego na prywatny). Uruchomienie tej funkcji sprawi, że odczyt/zapis danych będzie miał wpływ na odczyt/zapis bloków, pisanie słów ochronnych i operację zapisu numeru EPC.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[]        | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|---------------|-----------------------|-----|
| 0x06 | 0xXX | 0x3a                 | 0x00 | Typ<br>wkładu | LSB                   | MSB |

**InlayType:** typ tagu, prawidłowa wartość tego parametru to 0 ~ 254.

---

## Włącz sprawdzanie anteny

Funkcją tego polecenia jest włączanie/wyłączanie sprawdzania anteny podczas operacji odczytu/zapisu tagów.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[]   | Przełącznik |     |
|------|------|------|----------|-------------|-----|
|      |      |      | Szachant | CRC-16      |     |
| 0x05 | 0xXX | 0x66 | 0xXX     | LSB         | MSB |

**CheckAnt:** 1 bajt, przełącznik

sprawdzania anteny.

0x00 – wyłącz

sprawdzanie anteny;

0x01 – włącz

sprawdzanie anteny.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik |     |
|------|------|----------------------|------|--------|-------------|-----|
|      |      |                      |      |        | CRC-16      |     |
| 0x05 | 0xXX | 0x66                 | 0x00 | --     | LSB         | MSB |

## Modyfikowanie interfejsu komunikacyjnego

Funkcją tego polecenia jest skonfigurowanie interfejsu komunikacyjnego między RRU2881 a hostem. Ustawienie tego parametru zostanie zachowane nawet wtedy, gdy czytnik nie będzie miał dostępu do zasilania. Można wybrać tylko jeden typ interfejsu, użytkownik powinien podłączyć czytnik zgodnie z tą konfiguracją.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[]             | Przełącznik<br>CRC-16 |     |
|------|------|------|--------------------|-----------------------|-----|
|      |      |      | Typ<br>komunikacji |                       |     |
| 0x05 | 0xXX | 0x6A | 0xXX               | LSB                   | MSB |

CommType: 1 bajt, typ interfejsu komunikacyjnego. 0x00 – włącz połączenie USB; 0x01 – włącz połączenie UART.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

#### UWAGA:

1. Gdy połączenie USB jest włączone, połączenie UART nie będzie dostępne. Podobnie, gdy połączenie UART jest włączone, połączenie USB nie będzie dostępne.
2. Po modyfikacji interfejsu komunikacyjnego ustawienie nie będzie obowiązywać, dopóki czytnik nie zostanie prawidłowo wyłączony i ponownie uruchomiony.

#### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x6A                 | 0x00 | --     | LSB                   | MSB |

Modyfikowanie lub wczytywanie konfiguracji progu tłumienia odbicia anteny

Funkcją tego polecenia jest modyfikacja lub załadowanie progu tłumienia odbicia w celu sprawdzenia anteny. Domyślny próg tłumienia odbicia wynosi -6dB. Ustawienie tego parametru zostanie

zachowane nawet wtedy, gdy czytnik nie będzie miał dostępu do zasilania.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[]      | Przeказnik<br>CRC-16 |     |
|------|------|------|-------------|----------------------|-----|
|      |      |      | ZwrotStrata |                      |     |
| 0x05 | 0xXX | 0x6e | 0xXX        | LSB                  | MSB |

**ReturnLoss:** akcja na konfiguracji

Return loss. bit7: załaduj/zmodyfikuj flagę straty powrotnej.

0 – załaduj konfigurację

bieżącej strat odbiciowych; 1 –

Zmodyfikuj konfigurację tłumienia

odbicia. bit6 ~ bit0: konfiguracja

strat zwrotnych Dla bit7 = 0, bit6 ~

bit0 zostanie zignorowany.

Dla bit7 = 1, bit6 ~ bit0 jest pożądaną konfiguracją tłumienia odbicia.

Zakres tłumienia odbicia wynosi 0 ~ 20, zgodnie z 0dB ~ 20dB.

Inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[]      | Przeказnik<br>CRC-16 |     |
|------|------|----------------------|------|-------------|----------------------|-----|
|      |      |                      |      | ZwrotStrata |                      |     |
| 0x06 | 0xXX | 0x6e                 | 0x00 | 0xXX        | LSB                  | MSB |

**ReturnLoss:** bieżąca konfiguracja tłumienia

odbicia. bit7: zarezerwowana, domyślne

ustawienie to 0. bit6 ~ bit0: bieżący próg tłumienia

odbicia anteny. Zakres wynosi 0 ~ 20.

Modyfikowanie konfiguracji maksymalnej długości EPC/TID dla bufora pamięci

Funkcją tego polecenia jest modyfikacja maksymalnej długości EPC/TID dla inwentaryzacji z buforem pamięci. Ta modyfikacja spowoduje wyczyszczenie wszystkich danych tagów przechowywanych wcześniej w buforze pamięci.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[]        | Przeказnik<br>CRC-16 |     |
|------|------|------|---------------|----------------------|-----|
|      |      |      | Zapisz<br>Len |                      |     |
| 0x05 | 0xXX | 0x70 | 0xXX          | LSB                  | MSB |

**SaveLen:** 1 bajt, maksymalna długość EPC/TID.

0 – długość 128 bitów, czyli 16 bajtów. Maksymalna ilość tagów, które można przechowywać, wynosi 528 dla tego ustawienia; 1 – długość 496 bitów, czyli 62 bajty. Maksymalna ilość tagów, które można przechowywać, wynosi 160 dla tego ustawienia.

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|----------------------|------|--------|----------------------|-----|
| 0x05 | 0xXX | 0x70                 | 0x00 | --     | LSB                  | MSB |

załaduj konfigurację maksymalnej długości EPC/TID

Funkcją tego polecenia jest załadowanie konfiguracji maksymalnej długości EPC/TID dla bufora pamięci czytnika

**Ramka poleceń:**

| Len | Adr | Cmd | Dane[] | Przeказnik<br>CRC-16 |
|-----|-----|-----|--------|----------------------|
|-----|-----|-----|--------|----------------------|

|      |      |      |    |     |     |
|------|------|------|----|-----|-----|
| 0x04 | 0xXX | 0x71 | -- | LSB | MSB |
|------|------|------|----|-----|-----|

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[]        | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|---------------|-----------------------|-----|
| 0x06 | 0xXX | 0x71                 | 0x00 | Zapisz<br>Len | LSB                   | MSB |

**SaveLen:** 1 bajt, maksymalna  
długość EPC/TID. 0 –  
długość 128 bitów, czyli 16  
bajtów; 1 – długość 496  
bitów, czyli 62 bajty.

Uzyskiwanie danych z bufora pamięci

Funkcją tego polecenia jest uzyskanie wszystkich informacji o tagach z pamięci czytnika. To polecenie nie będzie miało wpływu na stan przechowywanych danych. Dostęp do danych można mieć wielokrotnie. Z drugiej strony, polecenie inwentaryzacji tagów (patrz rozdział 8.2.1) usunie wszystkie dane z bufora pamięci.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0x04 | 0xXX | 0x72 | --     | LSB                   | MSB |

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] |                               | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-------------------------------|-----------------------|-----|
|      |      |                      |      | Num    | Dane EPC                      |                       |     |
| 0xXX | 0xXX | 0x72                 | 0xXX | 0xXX   | EPC-1,<br>EPC-2, ...<br>EPC-n | LSB                   | MSB |

**Status:** status odpowiedzi, szczegółowy opis jest pokazany w poniższej tabeli

| Stan | Opis  |
|------|---|
| 0x01 | operacja zakończona, czytnik zgłosi wszystkie zapytane informacje o tagach do hosta                                       |
| 0x03 | Czytnik nie jest w stanie odpowiedzieć na wszystkie dane w jednej ramce, dalsze dane będą przesyłane w kolejnych ramkach. |

**Liczba:** 1 bajt, ilość EPC/TID zawarta w tej odpowiedzi.

**Dane EPC:** oznaczanie danych EPC/TID przechowywanych w buforze pamięci.

| EPC-n  |      |                    |                      |        |
|--------|------|--------------------|----------------------|--------|
| Mrówka | Len  | Świadcstwo EPC/TID | Wskaźnik RSSI (RSSI) | Hrabia |
| 0xXX   | 0xXX | N bajtów           | 0xXX                 | 0xXX   |

**Mrówka:** wskazuje, która antena wysłała zapytanie o tag. Na przykład 0x05 to 0101 w systemie binarnym, co oznacza Antenę 3 i Antenę 1.

**Len:** 1 bajt, długość bajtu EPC/TID.

**EPC / TID:** numer EPC / dane TID tagu, długość danych powinna być identyczna z Len. Każdy numer EPC tagu lub dane TID jest najbardziej znaczącym słowem i najbardziej znaczącym bajtem w słowie jako pierwsze.

**RSSI:** 1 bajt, siła sygnału czytnika w momencie zapytania o tag.

**Liczba:** łączna liczba wykrytych czasów dla określonego tagu.

0xFF – całkowity czas wykrywania jest większy lub równy 255.

---

## Wyczyść bufor pamięci

Funkcją tego polecenia jest wyczyszczenie wszystkich przechowywanych danych tagów z bufora pamięci.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|------|--------|----------------------|-----|
| 0x04 | 0xXX | 0x73 | --     | LSB                  | MSB |

### Ramka poleceń:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|----------------------|------|--------|----------------------|-----|
| 0x05 | 0xXX | 0x73                 | 0x00 | --     | LSB                  | MSB |

Uzyskaj łączną ilość tagów z bufora pamięci

Funkcją tego polecenia jest uzyskanie całkowitej ilości tagów przechowywanych w buforze pamięci. **Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|------|--------|----------------------|-----|
| 0x04 | 0xXX | 0x74 | --     | LSB                  | MSB |

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[]  | Przeказnik<br>CRC-16 |     |
|------|------|----------------------|------|---------|----------------------|-----|
|      |      |                      |      | Hrabia  |                      |     |
| 0x07 | 0xXX | 0x74                 | 0x00 | 2 bajty | LSB                  | MSB |

**Liczba:** 2 bajty, całkowita ilość znacznika przechowywanego w buforze pamięci, najpierw najbardziej znaczący bajt.

Modyfikacja parametrów trybu inwentaryzacji w czasie rzeczywistym

Funkcją tego polecenia jest modyfikowanie parametrów powiązanych z inwentaryzacją w czasie rzeczywistym.

---

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0xXX | 0xXX | 0x75 | --     | LSB                   | MSB |

**Parametry danych[]:**

| Dane[]      |              |                              |            |                   |        |
|-------------|--------------|------------------------------|------------|-------------------|--------|
| TagProtokół | Czas odczytu | FliterTime (Czas Określenia) | QValue     | Sesja             |        |
| 0xXX        | 0xXX         | 0xXX                         | 0xXX       | 0xXX              |        |
| Dane[]      |              |                              |            |                   |        |
| MaskMem     | MaskAdr      | MaskLen                      | Dane maski | AdrTID powiedział | LenTID |
| 0xXX        | 2Bajty       | 0xXX                         | zmienna    | 0xXX              | 0xXX   |

**TagProtocol:** 1 bajt, definicja typu tagu dla inwentaryzacji w czasie rzeczywistym.

- 0 – Znaczniki EPC C1G2 (ISO18000-6C).
- 1 – Znaczniki ISO18000-6B. Konfiguracje następujących parametrów zostaną zignorowane dla 6B, QValue, Session, MaskMem, MaskAdr, MaskLen, MaskData, AdrTID, LenTID.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**ReadPauseTime:** 1 bajt, przerwa czasowa między 2 inwentaryzacjami w czasie rzeczywistym.

0x00 – 10ms;

---

0x01 – 20ms;  
0x02 – 30ms;  
0x03 – 50ms;  
0x04 – 100ms.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**FliterTime:** 1 bajt, czas filtrowania tagów. Prawidłowa wartość tego parametru to 0 ~ 255, odpowiada  $(0 \sim 255) \cdot 1s$ . W inwentaryzacji w czasie rzeczywistym, jeśli czytnik wykryje określony tag więcej niż 1 raz, czytnik prześle informacje o tagu tego tagu tylko raz w określonym czasie filtrowania. Dla FliterTime = 0 wyłącz funkcję filtrowania tagów.

**QValue:** 1 bajt.

bit7: zarezerwowane, ustawione na 0.

bit6: Wskaźnik

strategii. 0 –  
strategia  
ogólna; 1 –  
Strategia  
specjalna.

bit5 ~ bit0: pierwotna wartość Q inwentarza tagów EPC, ustawienie wartości Q powinno być zgodne z regułą:  $2^Q \approx$  całkowitą liczbę tagów w efektywnym polu. Zakres wartości Q wynosi 0 ~ 15,

Jeśli w tym polu zostanie podana inna wartość, czytnik zwróci stan błędu parametru w ramce odpowiedzi.

**Sesja:** 1 bajt, wartość sesji inwentarza tagów EPC.

0x00 – zastosuj S0 jako wartość sesji;

---

0x01 – zastosuj S1 jako wartość sesji;  
0x02 – zastosuj S2 jako wartość sesji;  
0x03 – zastosuj S3 jako wartość sesji;  
0xff – zastosuj inteligentną konfigurację czytnika (dotyczy tylko w inwentarzu EPC).

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;  
0x02 – pamięć TID;  
0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskAdr:** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

**NUTA:**

1. Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną dostarczone, żaden wzór maski nie będzie uwzględniony w inwentarzu
2. W trybie inwentaryzacji w czasie rzeczywistym czytnik zastosuje predefiniowane parametry anteny i czasu inwentaryzacji do inwentaryzacji.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x75                 | 0x00 | --     | LSB                   | MSB |

Modyfikowanie trybu pracy czytnika

Funkcją tego polecenia jest modyfikacja trybu pracy czytnika.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[]          | Przełącznik<br>CRC-16 |     |
|------|------|------|-----------------|-----------------------|-----|
|      |      |      | Tryb<br>odczytu |                       |     |
| 0x05 | 0xXX | 0x76 | 0xXX            | LSB                   | MSB |

**ReadMode:** 1 bajt, bieżący tryb pracy czytnika.

0 – tryb odbierania;

1 – tryb inwentaryzacji w czasie rzeczywistym;

2 – Tryb inwentaryzacji w czasie rzeczywistym z wyzwalaczem. W przypadku tej konfiguracji czytnik będzie wykonywał inwentaryzację tylko wtedy, gdy poziom TTL portu GPII jest niski. Gdy poziom TTL GPII wzrośnie wysoko, czytnik przejdzie w tryb czuwania.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x76                 | 0x00 | --     | LSB                   | MSB |

---

**UWAGA:**

Ustawienie tego parametru zostanie zachowane nawet wtedy, gdy czytnik nie będzie zasilany, tzn. czytnik będzie stosował poprzednie ustawienie trybu pracy przy każdym ponownym uruchomieniu.

**Format wyjściowy w trybie inwentaryzacji w czasie rzeczywistym:**

Po ustawieniu czytnika w trybie inwentaryzacji w czasie rzeczywistym, czytnik przeprowadzi automatyczną inwentaryzację zgodnie z predefiniowaną konfiguracją.

Jeśli zapytany zostanie o tag, który pasuje do wstępnie zdefiniowanego warunku, czytnik automatycznie prześle dane w następującym formacie:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0xXX | 0xXX | 0xee                 | 0xXX | --     | LSB                   | MSB |

**Status:** status odpowiedzi, szczegółowy opis jest pokazany w poniższej tabeli

| Stan | Opis  |
|------|---|
| 0x00 | Wykryto odpowiedni tag, czytnik również zgłosi wykryte dane tagu.                               |
| 0x28 | Nie wykryto odpowiedniego tagu w przedziale czasu pakietu pulsu, czytnik przekaże pakiet pulsu. |

(1) Status odpowiedzi to 0x00, parametry danych [] są pokazane poniżej,

| Dane[] |      |                       |                            |
|--------|------|-----------------------|----------------------------|
| Mrówka | Len  | Świadectwo<br>EPC/TID | Wskaźnik<br>RSSI<br>(RSSI) |
| 0xXX   | 0xXX | N bajtów              | 0xXX                       |

---

Definicja  
parametrów

**Mrówka:** wskazuje, która antena wysłała zapytanie o tag. Na przykład 0x05 to 0101 w systemie binarnym, co oznacza Antenę 3 i Antenę 1.

**Len:** 1 bajt, długość bajtu EPC/TID.

**EPC / TID:** numer EPC / dane TID tagu, długość danych powinna być identyczna z Len. Jednostką danych EPC C1 G2 jest słowo. Każdy numer EPC tagu lub dane TID jest najbardziej znaczącym słowem i najbardziej znaczącym bajtem w słowie jako pierwsze.

**RSSI:** 1 bajt, siła sygnału czytnika w momencie zapytania o tag.

(2) Status odpowiedzi to 0x28, parametry danych [] są pokazane poniżej

| Dane[]     |                       |               |
|------------|-----------------------|---------------|
| Nr pakietu | AntStatus (AntStatus) | Łączna liczba |
| 4 bajty    | 4 bajty               | 4 bajty       |

Definicja  
parametrów

**PacketNo:** numer seryjny pulsu. 4 bajty, najpierw bajt najbardziej znaczący.

Numer seryjny będzie się zwiększał dla każdej operacji przesyłania pakietów pulsu. Jeśli liczba seryjna pulsu przekroczy maksymalną wartość 4 bajtów, numer seryjny zostanie ponownie przeliczony od 0.

**AntStatus:** aktualny stan 4 anten, kolejkowanych od 1 do 4 zgodnie z kolejnością bajtów.

0x00 – antena docelowa jest bezczynna,  
nie jest to skuteczna antena

inwentaryzacyjna 0x01 – antena docelowa

---

działa poprawnie; 0x02 – antena docelowa jest odłączona.

**TotalCount:** łączna ilość tagów, o które zapytano od początku bieżącej automatycznej inwentaryzacji. Wielokrotne identyfikowanie tagu spowoduje również zwiększenie wartości TotalCount. 4 bajty, z najbardziej znaczącym bajtem na początku. Jeśli łączna ilość żądanego tagu przekroczy maksymalną wartość 4 bajtów, TotalCount przeliczy ponownie od 0.

W trybie inwentaryzacji w czasie rzeczywistym czytnik odpowie tylko na 3 polecenia:

1. uzyskać informacje o czytniku (patrz rozdział 8.4.1);
2. zmodyfikuj tryb pracy czytnika (patrz rozdział 8.4.22);
3. Uzyskaj parametry trybu automatycznego (patrz rozdział 8.4.23).

Uzyskaj tryb pracy czytnika

Zadaniem tego polecenia jest uzyskanie parametrów trybu pracy czytnika.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[] | Przeказnik CRC-16 |     |
|------|------|------|--------|-------------------|-----|
| 0x04 | 0xXX | 0x77 | --     | LSB               | MSB |

**Ramka odpowiedzi:**

| Len | Adr | reCmd<br>powiedział<br>: | Stan | Dane[] | Przeказnik<br>k CRC-16 |
|-----|-----|--------------------------|------|--------|------------------------|
|-----|-----|--------------------------|------|--------|------------------------|

|          |          |      |          |  |     |         |
|----------|----------|------|----------|--|-----|---------|
| 0xX<br>X | 0xX<br>X | 0x77 | 0x0<br>0 | ReadMode, TagProtocol<br>,<br>ReadPauseTime,<br>FilterTime, QValue,<br>Session, MaskMem,<br>MaskAdr,<br>MaskLen, MaskData,<br>AdrTID, LenTID | LSB | MS<br>B |
|----------|----------|------|----------|--|-----|---------|

Definicja parametrów

**ReadMode:** 1 bajt, bieżący tryb pracy czytnika.

**TagProtocol:** 1 bajt, definicja typu tagu dla inwentaryzacji w czasie rzeczywistym.

**ReadPauseTime:** 1 bajt, przerwa czasowa między 2 inwentaryzacjami w czasie rzeczywistym.

**FilterTime:** 1 bajt, czas filtrowania tagów w inwentaryzacji w czasie rzeczywistym.

**QValue:** 1 bajt.

bit7: zarezerwowane, ustawione na 0.

bit6: Wskaźnik strategii. 0 – strategia ogólna; 1 – Strategia specjalna.

bit5 ~ bit0: pierwotna wartość Q inwentarza tagów EPC, ustawienie wartości Q powinno być zgodne z regułą:  $2^Q \approx$  całkowitą liczbę tagów w efektywnym polu. Zakres wartości Q wynosi 0

---

~ 15, jeśli w tym polu zostanie dostarczona inna wartość, czytnik zwróci stan błędu parametru w ramce odpowiedzi.

**Sesja:** 1 bajt, wartość sesji w trybie czasu rzeczywistego dla inwentaryzacji EPC.

**MaskMem, MaskAdr, MaskLen, MaskData:** stan maski dla inwentaryzacji tagów EPC C1G2 w trybie czasu rzeczywistego. MaskMem i MaskLen mają odpowiednio 1 bajt. MaskAdr ma długość 2 bajtów, najpierw najbardziej znaczący bajt. MaskData ma stałą długość 32 bajtów, wypełnij zawartość wartości MaskLen zerem.

**AdrTID:** początkowy adres słowny inwentarza pamięci TID.

**LenTID:** długość danych dla operacji inwentaryzacji TID.

Załaduj/zmodyfikuj przerwę w czasie pakietów pulsu w inwentaryzacji w czasie rzeczywistym

Funkcją tego polecenia jest modyfikowanie przerwy czasowej przekazywania pakietów pulsu w trybie inwentaryzacji w czasie rzeczywistym. Dla HeartBeatTime = 0 wyłącz dostarczanie pakietów heartbeat.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[]                 | Przełącznik<br>CRC-16 |     |
|------|------|------|------------------------|-----------------------|-----|
|      |      |      | Czas<br>bicia<br>serca |                       |     |
| 0x05 | 0xXX | 0x78 | 0xXX                   | LSB                   | MSB |

**HeartBeatTime:** przerwa czasowa między dostarczeniem 2 pakietów pulsu bit7: akcja przy konfiguracji przerwy czasowej pulsu 0 – załaduj konfigurację przerwy czasowej pulsu;

---

1 – Zmodyfikuj konfigurację przerwy czasowej pulsu.

bit6 ~ bit0: konfiguracja przerwy

czasowej bicia serca Dla bit7 = 0,

bit6 ~ bit0 zostanie zignorowany.

Dla bit7 = 1, bit6 ~ bit0 to konfiguracja przerwy czasowej między 2

dostawami pakietów pulsu Dla HeartBeatTime = 0 wyłącz

dostarczanie pakietów heartbeat.

#### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[]                 | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|------------------------|-----------------------|-----|
|      |      |                      |      | Czas<br>bicia<br>serca |                       |     |
| 0x06 | 0xXX | 0x78                 | 0x00 | 0xXX                   | LSB                   | MSB |

**HeartBeatTime:** bieżąca konfiguracja podziału czasu

pakietu heartbeat. bit7: zarezerwowana, wartość

domyślna to 0.

bit6 ~ bit0: bieżąca konfiguracja przerwy czasowej pakietu heartbeat.

Prawidłowa wartość tego parametru to 0 ~ 127, odpowiada (0 ~

127)\*30s.

#### UWAGA:

Czytnik dostarczy pakiet pulsu tylko wtedy, gdy żaden tag nie

zostanie wykryty przez kolejny okres (dłuższy niż przerwa w czasie

bicia serca)

Modyfikowanie konfiguracji mocy RF oddzielnie dla operacji zapisu  
 Funkcją tego polecenia jest modyfikowanie mocy RF dla operacji zapisu. Domyślna wartość parametru power to 0x00. Ustawienie tego parametru zostanie zachowane nawet wtedy, gdy czytnik nie będzie miał dostępu do zasilania.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[]     | Przełącznik<br>CRC-16 |     |
|------|------|------|------------|-----------------------|-----|
|      |      |      | Moc zapisu |                       |     |
| 0x05 | 0xXX | 0x79 | 0xXX       | LSB                   | MSB |

**WritePower:** konfiguracja mocy.

bit7: włącz/wyłącz oddzielną konfigurację zasilania RF dla operacji zapisu.

- 0 – wyłącz oddzielną konfigurację zasilania RF, moc będzie identyczna zarówno dla poleceń odczytu, jak i zapisu.
- 1 – włącz oddzielną konfigurację zasilania RF. Prawidłowa wartość mocy RF wynosi 0 ~ 30, dla ustawienia 30 moc wyjściowa wynosi około 1 W.

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x79                 | 0x00 | --     | LSB                   | MSB |

Oddzielna moc RF będzie obowiązywać dla następujących poleceń:

**Polecenia EPC C1G2 (ISO18000-6C)**

- polecenie zapisu danych;
- wpisz polecenie numeru EPC;
- Rozszerzone polecenie zapisu danych z przypisaną maską;
- Polecenie kasowania bloku; Polecenie pisania bloków.

---

## Polecenie ISO18000-6B

- Polecenie zapisu danych

Załaduj konfigurację mocy RF operacji zapisu

Funkcją tego polecenia jest załadowanie predefiniowanej oddzielnej konfiguracji mocy RF operacji zapisu.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0x04 | 0xXX | 0x7A | --     | LSB                   | MSB |

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[]        | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|---------------|-----------------------|-----|
|      |      |                      |      | Moc<br>zapisu |                       |     |
| 0x06 | 0xXX | 0x7A                 | 0x00 | 0xXX          | LSB                   | MSB |

**WritePower:** konfiguracja mocy.

bit7: konfiguracja oddzielnej konfiguracji zasilania RF dla operacji zapisu.

- 0 – wyłączono oddzielną konfigurację zasilania RF, moc będzie identyczna zarówno dla poleceń odczytu, jak i zapisu.
- 1 – włączono oddzielną konfigurację zasilania RF. Prawidłowa wartość mocy RF wynosi 0 ~ 30, dla ustawienia 30 moc wyjściowa wynosi około 1 W.

Modyfikowanie lub ładowanie konfiguracji maksymalnego czasu ponawiania zapisu

Funkcją tego polecenia jest modyfikowanie lub ładowanie konfiguracji maksymalnego czasu ponawiania zapisu. W przypadku

zapisywania tagów i skojarzonych poleceń, jeśli wystąpił błąd, czytnik ponowi próbę wykonania operacji aż do powodzenia lub osiągnięcia maksymalnego czasu ponawiania zapisu. Domyślna wartość czasu ponawiania próby to 3. Ustawienie tego parametru zostanie zachowane nawet wtedy, gdy czytnik nie będzie miał dostępu do zasilania.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[]             | Przełącznik<br>CRC-16 |     |
|------|------|------|--------------------|-----------------------|-----|
|      |      |      | Ponawianie<br>prób |                       |     |
| 0x05 | 0xXX | 0x7B | 0xXX               | LSB                   | MSB |

**RetryTimes:** akcja przy konfiguracji

maksymalnego czasu ponawiania zapisu bit7:

załaduj/zmodyfikuj konfigurację

maksymalnego czasu ponawiania zapisu

0 – załaduj maksymalny czas ponawiania próby zapisu;

1 – Zmodyfikuj maksymalny czas ponawiania prób zapisu; bit6 ~ bit3: zarezerwowane.

bit2 ~ bit0: maksymalny

czas ponawiania zapisu Dla

bit7 = 0, bit6 ~ bit0 zostanie

zignorowany.

Dla bit7 = 1, bit2 ~ bit0 to konfiguracja maksymalnego czasu ponawiania próby. Zakres tego parametru wynosi 0 ~ 7.

### Ramka odpowiedzi:

| Len | Adr |  | Stan | Dane[] |  |
|-----|-----|--|------|--------|--|
|     |     |  |      |        |  |

|      |      |                      |      |                    |                       |     |
|------|------|----------------------|------|--------------------|-----------------------|-----|
|      |      | reCmd<br>powiedział: |      | Ponawianie<br>prób | Przełącznik<br>CRC-16 |     |
| 0x06 | 0xXX | 0x7B                 | 0x00 | 0xXX               | LSB                   | MSB |

Definicja parametru

**RetryTimes:** wartość maksymalnego czasu ponawiania zapisu bit7 ~ bit3: zarezerwowana, wartość domyślna to 0.  
bit2 ~ bit0: ustawienie maksymalnego czasu ponawiania zapisu. Prawidłowy zakres czasu ponawiania wynosi 0 ~ 7.

Licznik czasu ponawiania prób zacznie obowiązywać dla następującego polecenia:

#### Polecenia EPC C1G2 (ISO18000-6C)

- polecenie zapisu danych;
- wpisz polecenie numeru EPC;
- Rozszerzone polecenie zapisu danych z przypisaną maską;
- Polecenie kasowania bloku;
- blokuj polecenie pisania.

#### Polecenie ISO18000-6B

- Polecenie zapisu danych

Modyfikowanie hasła funkcji niestandardowych tagów

Funkcją tego polecenia jest modyfikacja hasła dostępu do funkcji dostosowanych do tagów. To hasło zacznie obowiązywać po uruchomieniu funkcji Peek tagu Monza4QT. Ustawienie tego parametru zostanie zachowane nawet wtedy, gdy czytnik nie będzie miał dostępu do zasilania.

**Ramka poleceń:**

|     |     |     |        |  |
|-----|-----|-----|--------|--|
| Len | Adr | Cmd | Dane[] |  |
|-----|-----|-----|--------|--|

|      |      |      | UżytkownikPwd | Przeказnik<br>CRC-16 |     |
|------|------|------|---------------|----------------------|-----|
| 0x08 | 0xXX | 0x7D | 4 bajty       | LSB                  | MSB |

**UserPwd:** 4 bajty, najpierw najbardziej znaczący bajt. Hasło dostępu dostosowane do potrzeb użytkownika.

#### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|----------------------|------|--------|----------------------|-----|
| 0x05 | 0xXX | 0x7D                 | 0x00 | --     | LSB                  | MSB |

Uzyskaj hasło do funkcji niestandardowych tagów

Funkcją tego polecenia jest uzyskanie hasła dostępu do funkcji dostosowanych do tagów.

#### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|------|--------|----------------------|-----|
| 0x04 | 0xXX | 0x7E | --     | LSB                  | MSB |

#### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[]        | Przeказnik<br>CRC-16 |     |
|------|------|----------------------|------|---------------|----------------------|-----|
|      |      |                      |      | UżytkownikPwd |                      |     |
| 0x09 | 0xXX | 0x7E                 | 0x00 | 4 bajty       | LSB                  | MSB |

**UserPwd:** 4 bajty, najpierw najbardziej znaczący bajt. Hasło dostępu dostosowane do potrzeb użytkownika.

Załaduj/zmodyfikuj profil czytelnika

Funkcją tego polecenia jest modyfikacja / wczytanie konfiguracji profilu wewnętrznego czytelnika. Domyślna wartość profilu to 1.

Ustawienie tego parametru zostanie zachowane nawet wtedy, gdy czytnik nie będzie miał dostępu do zasilania.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przełącznik |     |
|------|------|------|--------|-------------|-----|
|      |      |      | profil | CRC-16      |     |
| 0x05 | 0xXX | 0x7F | 0xXX   | LSB         | MSB |

### Profil definicji parametru:

konfiguracja numeru profilu.

Bit7: Akcja na konfiguracji

numeru profilu. 0 – numer profilu obciążenia;

1 – Zmień numer profilu.

bit6 ~ bit0: konfiguracja numeru profilu.

Dla bit7 = 0, bit6 ~ bit0 zostanie zignorowany.

Dla bit7 = 1, bit6 ~ bit0 to nowa konfiguracja numeru profilu.

Prawidłowy zakres tego parametru to 0 ~ 3.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd powiedział: | Stan | Dane[] | Przełącznik |     |
|------|------|-------------------|------|--------|-------------|-----|
|      |      |                   |      | profil | CRC-16      |     |
| 0x06 | 0xXX | 0x7F              | 0x00 | 0xXX   | LSB         | MSB |

Profil definicji parametru: bieżąca konfiguracja numeru

profilu. bit7: zarezerwowany, wartość domyślna to 0. bit6 ~

bit0: bieżący numer profilu czytnika. Prawidłowy zakres tego parametru to 0 ~ 3.

---

## Konfiguracja różnych profili

Profil0: Tari 25uS, FM0 40KHz;

Profil1: Tari 25uS, Miller 4 250 kHz (zalecana konfiguracja, domyślne ustawienie systemu) Profil 2: Tari 25uS, Miller 4 300 kHz; Profil 3: Tari 6.25uS, FM0 400KHz.

Synchronizuj znacznik czasu EM4325

Funkcją tego polecenia jest rozgłaszanie znacznika czasu do wszystkich tagów EM4325 w efektywnym polu.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[]      | Przeказnik<br>CRC-16 |     |
|------|------|------|-------------|----------------------|-----|
|      |      |      | Czas<br>UTC |                      |     |
| 0x08 | 0xXX | 0x85 | 4 bajty     | LSB                  | MSB |

**Czas UTC:** 4 bajty, znacznik czasu UTC do emisji, najpierw najbardziej znaczący bajt. Jest to druga kwota od 1970-01-01 00:00:00.

Synchronizacja zakończy się niepowodzeniem, jeśli MSB czasu UTC ma wartość 0.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|----------------------|------|--------|----------------------|-----|
| 0x05 | 0xXX | 0x85                 | 0x00 | —      | LSB                  | MSB |

### UWAGA:

1. Tag EM4325 musi pozostać w trybie BAP, aby odebrać i zsynchronizować znacznik czasu.
2. Synchronizacja zakończy się niepowodzeniem, jeśli dostarczony MSB czasu UTC ma wartość 0.

3. Synchronizacja zakończy się niepowodzeniem, jeśli zostanie ustawiona jakakolwiek flaga alertu. Flaga alertu może być usunięta za pomocą polecenia "Zresetuj alert EM4325" lub tymczasowo wyłączyć funkcje alertów poprzez modyfikację powiązanych rejestrów EM4325.
4. Tagi EM4325 nie będą generować odpowiedzi na polecenie synchronizacji znacznika czasu. W związku z tym czytnik zgłosi pomyślny stan hosta po dostarczeniu polecenia synchronizacji znacznika czasu do tagu.
5. Po skonfigurowaniu wewnętrznego rejestru czasu UTC EM4325 rejestr ten będzie zwiększał się o 1 na sekundę. Ten licznik czasu zatrzyma się, jeśli 4-bajtowy rejestr przepełnienia 0xFFFFFFFF i z powrotem do 0 lub powiązany rejestr konfiguracji systemu w obszarze użytkownika zostanie zmodyfikowany.

Uzyskaj dane o temperaturze EM4325

Funkcją tego polecenia jest uzyskanie danych o temperaturze z tagu EM4325.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|------|--------|----------------------|-----|
| 0xXX | 0xXX | 0x86 | --     | LSB                  | MSB |

**Parametry danych[]:**

| Dane[] |  |   |                           |     |
|--------|--|---|---------------------------|-----|
| ENum   | Świadectwo<br>charakterystyki<br>energetycznej | SendUID<br>(Wyślij<br>identyfikator<br>SendUID) | NewSample<br>(NowyPróbka) | Pwd |

|         |         |         |            |            |
|---------|---------|---------|------------|------------|
| 0xXX    | zmienna | 0xXX    | 0xXX       | 4<br>bajty |
| MaskMem | MaskAdr | MaskLen | Dane maski |            |
| 0xXX    | 2Bajty  | 0xXX    | zmienna    |            |

**ENum:**

0x00 ~ 0x0f – ENum oznacza długość liczby EPC. Jednostką ENum jest słowo, a długość EPC powinna być mniejsza niż 15 słów. W takiej sytuacji parametry MaskMen, MaskAdr, MaskLen, MaskData powinny pozostać puste.

0xff – Zamiast parametrów EPC należy podać MaskMem, MaskAdr, MaskLen, MaskData. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**EPC:** numer EPC znacznika docelowego. Długość świadectwa charakterystyki energetycznej zależy od podanego numeru EPC. Jednostką numeru EPC jest słowo. Podczas komunikacji najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie. Niezbędne jest podanie PEŁNEGO EPC w tym poleceniu.

**SendUID:** 1 bajt, flaga odpowiedzi UID. 0x00 – nie odpowiadaj UID; 0x01 – UID odpowiedzi.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**NewSample:** 1 bajt, nowa flaga danych temperatury. 0x00 – uzyskanie poprzednich danych

---

temperatury; 0x01 – uzyskanie nowych danych temperatury.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to górna liczba bajtów hasła dostępu, a ostatnie dwa bajty to dolne bajty. Najbardziej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najbardziej znaczący bit w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczącym bitem 32-bitowego hasła dostępu tagu jest najmniej znaczący bit w 4. (ostatnim) bajcie Pwd. Psw powinno być ustawione na all-zero lub ustawione na poprawne hasło dostępu.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć TID;

0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskAdr:** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa  $\text{MaskLen}/8$ . Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

**NUTA:**

Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną dostarczone, pełny numer EPC zostanie użyty jako wzór maski w ekwipunku.

### Ramka odpowiedzi:

| Len      | Adr      | reCmd<br>powiedzi<br>ał: | Sta<br>n | Dane[]      |                        |            | Przekaźn<br>ik CRC-<br>16 |     |
|----------|----------|--------------------------|----------|-------------|------------------------|------------|---------------------------|-----|
|          |          |                          |          | UID         | Dane<br>SenserDa<br>ta | UTC        |                           |     |
| 0xX<br>X | 0xX<br>X | 0x86                     | 0x0<br>0 | zmienn<br>a | 4Bajty                 | 4Bajt<br>y | LSB                       | MSB |

**UID:** unikalny numer identyfikacyjny znacznika EM4325.

W przypadku SendUID = 0 w odpowiedzi nie zostanie dostarczone żadne pole UID;

W przypadku SendUID = 1 pole UID zostanie dostarczone w odpowiedzi, długość pola zależy od tagu, dla różnych odpowiedzi tagów długość UID może się zmieniać między 8 bajtami, 10 bajtami po 12 bajtów.

**SenserData:** dane o temperaturze zwrócone z tagu EM4325, długość danych wynosi 4 bajty, najpierw najbardziej znaczący bajt. Szczegółowy opis można znaleźć w definicji rejestru danych czujnika w arkuszu danych EM4325.

**UTC:** 4 bajty, bieżący znacznik czasu UTC znacznika, najpierw najbardziej znaczący bajt. Jest to druga kwota od 1970-01-01 00:00:00. Jeśli tag docelowy nie został zsynchronizowany lub znacznik czasu UTC tagu nie został skonfigurowany, zwracana wartość będzie równa zero.

Uzyskiwanie danych zewnętrznych za pośrednictwem EM4325 SPI

Funkcją tego polecenia jest użycie znacznika EM4325 jako urządzenia głównego SPI i uzyskanie danych z podłączonego do niego urządzenia podrzędnego SPI.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|------|--------|----------------------|-----|
| 0xXX | 0xXX | 0x87 | --     | LSB                  | MSB |

### Parametry danych[]:

| Dane<br>[]               |         |  |                            |           |                                   |                        |
|--------------------------|---------|--|----------------------------|-----------|-----------------------------------|------------------------|
| SPICmdRozmiar            | ENum    | Świadectwo charakterystyki energetycznej | SPIResSize (Rozmiar Wyspy) | SPISclock | SPIInitDelay (Opóźnienie sygnału) | SPIInterval powieział: |
| 0xXX                     | 0xXX    | zmienna                                  | 0xXX                       | 0xXX      | 0xXX                              | 0xXX                   |
| SPECYFIKA CJA SPECJALNA. | Pwd     | MaskMem                                  | MaskAdr                    | MaskLen   | Dane maski                        |                        |
| zmienna                  | 4 bajty | 0xXX                                     | 2 bajty                    | 0xXX      | zmienna                           |                        |

### ENum:

0x00 ~ 0x0f – ENum oznacza długość liczby EPC. Jednostką ENum jest słowo, a długość EPC powinna być mniejsza niż 15 słów. W

---

takiej sytuacji parametry MaskMem, MaskAdr, MaskLen, MaskData powinny pozostać puste.

0xff – Zamiast parametrów EPC należy podać MaskMem, MaskAdr, MaskLen, MaskData. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**EPC:** numer EPC znacznika docelowego. Długość świadectwa charakterystyki energetycznej zależy od podanego numeru EPC. Jednostką numeru EPC jest słowo. Podczas komunikacji najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie. Niezbędne jest podanie PEŁNEGO EPC w tym poleceniu.

**SPICmdSize:** 1 bajt, długość bajtu polecenia SPI. Prawidłowy zakres SPICmdSize to 1 ~ 8. Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**SPIResSize:** 1 bajt, długość bajtu odpowiedzi z urządzenia podrzędnego SPI. Prawidłowy zakres SPIResSize to 0 ~ 7. SPIResSize = 0 oznacza, że żadna odpowiedź nie zostanie wygenerowana. Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**SPISclk:** 1 bajt, zegar magistrali SPI.

0 – 40 kHz;

1 – 80 kHz;

2 – 160 KHz; 3 – 320 kHz.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**SPIInitDelay:** 1 bajt, opóźnienie dla inicjalizacji zegara SPI.

---

0 – 1 SCLK;

1 – 50 lat;

2 – 500us; 3 – 5 ms.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**SPIInterval:** 1 bajt, przerwa czasowa między każdym bajtem danych SPI.

0 – brak opóźnień;

1 – 50 lat;

2 – 100us; 3 – 500us.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**SPICmd:** dane polecenia SPI, które mają zostać dostarczone, długość danych musi być identyczna z wartością określoną w SPICmdSize.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to górna liczba bajtów hasła dostępu, a ostatnie dwa bajty to dolne bajty. Najbardziej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najbardziej znaczący bit w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczącym bitem 32-bitowego hasła dostępu tagu jest najmniej znaczący bit w 4. (ostatnim) bajcie Pwd. Psw powinno być ustawione na all-zero lub ustawione na poprawne hasło dostępu.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć TID;

0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskAdr:** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

**UWAGA:**

Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną dostarczone, pełny numer EPC zostanie użyty jako wzór maski w ekwipunku.

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[]             | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------------------|-----------------------|-----|
|      |      |                      |      | Dane<br>SPIResData |                       |     |
| 0xXX | 0xXX | 0x87                 | 0x00 | zmienna            | LSB                   | MSB |

**SPIResData:** dane zwrócone z urządzenia podrzędnego EM4325 SPI, prawidłowy zakres danych to 0 ~ 7 bajtów.

Długość SPIResData powinna być identyczna z wartością określoną w SPIResSize.

**UWAGA:**

Przed uzyskaniem danych zewnętrznych za pośrednictwem EM4325 SPI należy najpierw skonfigurować rejestr bajtów kontrolnych I/O EM4325 (I/O Control Word). Użytkownik powinien włączyć funkcję SPI, skonfigurować port SPI i włączyć rejestr podciągania pinów SPI.

---

## Zresetuj alert EM4325

Funkcją tego polecenia jest zresetowanie całej flagi ostrzegawczej tagu EM4325. Zegar UTC i funkcje monitorowania będą jasne.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[] | Przeказnik<br>CRC-16 |     |
|------|------|------|--------|----------------------|-----|
| 0xXX | 0xXX | 0x88 | --     | LSB                  | MSB |

### Parametry danych[]:

| Dane[] |   |                |             |             |             |               |
|--------|---|----------------|-------------|-------------|-------------|---------------|
| ENum   | Świadectwo<br>charakterysty<br>ki<br>energetyczne | Pw<br>d        | MaskMe<br>m | MaskAd<br>r | MaskLe<br>n | Dane<br>maski |
| 0xXX   | zmienna   | 4<br>bajt<br>y | 0xXX        | 2 bajty     | 0xXX        | zmienn<br>a   |

### ENum:

0x00 ~ 0x0f – ENum oznacza długość liczby EPC. Jednostką ENum jest słowo, a długość EPC powinna być mniejsza niż 15 słów. W takiej sytuacji parametry MaskMem, MaskAdr, MaskLen, MaskData powinny pozostać puste.

0xff – Zamiast parametrów EPC należy podać MaskMem, MaskAdr, MaskLen, MaskData. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**EPC:** numer EPC znacznika docelowego. Długość świadectwa charakterystyki energetycznej zależy od podanego numeru EPC.

---

Jednostką numeru EPC jest słowo. Podczas komunikacji najpierw zostanie przesłane najbardziej znaczące słowo i najbardziej znaczący bajt w słowie. Niezbędne jest podanie PEŁNEGO EPC w tym poleceniu.

**Pwd:** 4 bajty, jest to 4-bajtowe hasło dostępu do tagu. Od lewej do prawej, pierwsze dwa bajty to górna liczba bajtów hasła dostępu, a ostatnie dwa bajty to dolne bajty. Najbardziej znaczącym bitem 32-bitowego hasła dostępu do tagu jest najbardziej znaczący bit w 1. bajcie Pwd (od lewej do prawej). Podobnie, najmniej znaczącym bitem 32-bitowego hasła dostępu tagu jest najmniej znaczący bit w 4. (ostatnim) bajcie Pwd. Psw powinno być ustawione na all-zero lub ustawione na poprawne hasło dostępu.

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć TID;

0x03 – Pamięć użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskAdr:** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa MaskLen/8. Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

**UWAGA:**

Jeśli MaskMem, MaskAdr, MaskLen i MaskData nie zostaną dostarczone, pełny numer EPC zostanie użyty jako wzór maski w ekwipunku.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x88                 | 0x00 | --     | LSB                   | MSB |

### UWAGA:

Przed użyciem polecenia "zresetuj alert EM4325" należy najpierw skonfigurować bajt sterujący czujnika temperatury EM4325I (Temp Sensor Control Word 1), aby włączyć funkcję resetowania alertu EM4325; w przeciwnym razie tag EM4325 nie wygeneruje żadnej odpowiedzi na to polecenie.

### Modyfikowanie lub wczytywanie konfiguracji DRM

Funkcją tego polecenia jest modyfikowanie lub ładowanie trybu przełącznika DRM. Ustawienie tych parametrów zostanie utracone podczas wyłączenia zasilania. DRM będzie resetowany do stanu wyłączonego przy każdym uruchomieniu czytnika.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Dane[]      | Przełącznik<br>CRC-16 |     |
|------|------|------|-------------|-----------------------|-----|
|      |      |      | Kod<br>DRMM |                       |     |
| 0x05 | 0xXX | 0x90 | 0xXX        | LSB                   | MSB |

**DRMMode:** akcja na konfiguracji

DRM. bit7: załaduj/zmodyfikuj

flagę DRM.

0 – załaduj aktualną konfigurację DRM; 1 – zmodyfikuj konfigurację DRM.  
 bit6 ~ bit0: konfiguracja DRM  
 Dla bit7 = 0, bit6 ~ bit0 zostanie zignorowany.  
 Dla bit7 = 1, bit6 ~ bit0 jest pożądaną konfiguracją DRM. 0 – Wyłącz DRM; 1 – Włącz DRM.

Inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

#### Ramka odpowiedzi:

| Len  | Adr  | reCmd powieział: | Stan | Dane[]   | Przełącznik CRC-16 |     |
|------|------|------------------|------|----------|--------------------|-----|
|      |      |                  |      | Kod DRMM |                    |     |
| 0x06 | 0xXX | 0x90             | 0x00 | 0xXX     | LSB                | MSB |

**DRMMode:** akcja na konfiguracji

DRM. bit7: zarezerwowany, domyślne ustawienie to 0.

bit6 ~ bit0: bieżąca konfiguracja DRM  
 0 – DRM jest wyłączony; 1 – DRM jest włączony.

#### Zmierz tłumienie odbicia anteny

Funkcją tego polecenia jest pomiar tłumienia odbicia anteny w określonym punkcie częstotliwości.

### Ramka poleceń:

| Len  | Adr  | Cmd  | Datownik $\gamma$ [ ]    |        | Przeakaźnik<br>k CRC-16 |     |
|------|------|------|--------------------------|--------|-------------------------|-----|
|      |      |      | Częstotliwość<br>testowa | Mrówka |                         |     |
| 0x09 | 0xXX | 0x91 | 4 bajty                  | 0xXX   | LSB                     | MSB |

Definicja

parametrów

**TestFreq:** docelowy punkt częstotliwości dla pomiaru tłumienia odbicia. Jednostką TestFreq jest KHz, najpierw najbardziej znaczący bajt. Prawidłowy punkt częstotliwości musi być wielokrotnością 125 kHz lub 100 kHz. Czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**Ant:** 1 bajt, antena docelowa do tego pomiaru

0 – antena 1;

1 – antena 2;

2 – antena 3; 3 – antena 4.

Inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

### Ramka odpowiedzi:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane [ ]    | Przeakaźnik<br>CRC-16 |     |
|------|------|----------------------|------|-------------|-----------------------|-----|
|      |      |                      |      | ZwrotStrata |                       |     |
| 0x06 | 0xXX | 0x91                 | 0x00 | 0xXX        | LSB                   | MSB |

**ReturnLoss:** wynik pomiaru strat odbiciowych, jednostką ReturnLoss jest dB.

---

Zmierz aktualną temperaturę czynnika

Funkcją tego polecenia jest uzyskanie aktualnej temperatury czynnika.

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0x04 | 0xXX | 0x92 | --     | LSB                   | MSB |

**Ramka odpowiedzi:**

| Len      | Adr      | reCmd<br>powiedział<br>: | Stan     | Dane[]                            |                   | Przełącznik<br>CRC-16 |         |
|----------|----------|--------------------------|----------|-----------------------------------|-------------------|-----------------------|---------|
|          |          |                          |          | PlusMinus<br>(Dodatkowy<br>Minus) | Najwyższa<br>temp |                       |         |
| 0x0<br>7 | 0xX<br>X | 0x92                     | 0x0<br>0 | 0xXX                              | 0xXX              | LS<br>B               | MS<br>B |

**PlusMinus:** 1 bajt, zakres temperatur. 0 – poniżej 0 °C; 1 – powyżej 0 °C.

**Temp:** 1 bajt, wartość temperatury, jednostką jest °C.

**UWAGA:**

To polecenie jest obsługiwane tylko przez wbudowany czujnik czynnika temperatury.

Ustawianie parametrów rozszerzonych

Komputer hosta służy do konfiguracji różnych rozszerzonych parametrów czynnika, obecnie obsługuje tylko serię Ex10. **Ramka poleceń:**

| Len    | Adr  | Cmd  | Dane[] |                     |                          | Przełącznik<br>CRC-16 |     |
|--------|------|------|--------|---------------------|--------------------------|-----------------------|-----|
| 0x06+n | 0xXX | 0xEA | Flaga  | Certyfikat<br>CFGNo | CFGData<br>(n<br>bajtów) | LSB                   | MSB |

**Flaga:** 1 bajt, 0 - zapisz po wyłączeniu zasilania; 1 - nie zapisz po wyłączeniu zasilania;

**CFGNo:** 1 bajt, Służy do określania numeru sekwencyjnego parametru, który ma zostać ustawiony w bloku danych polecenia.

**CFGData:** Zawartość parametru, zobacz **Konfiguracja parametrów czytnika**, aby uzyskać szczegółowe informacje.

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0xEA                 | 0x00 | --     | LSB                   | MSB |

Ustawianie parametrów rozszerzonych

Komputer hosta służy do uzyskiwania rozszerzonych informacji o parametrach czytnika, obecnie obsługuje tylko serię Ex10

**Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[]              | Przełącznik<br>CRC-16 |     |
|------|------|------|---------------------|-----------------------|-----|
|      |      |      | Certyfikat<br>CFGNo |                       |     |
| 0x05 | 0xXX | 0xEB | 0xXX                | LSB                   | MSB |

**CFGNo:** 1 bajt, Służy do określania numeru sekwencyjnego parametru, który ma zostać ustawiony w bloku danych polecenia.

**Ramka odpowiedzi:**

| Len    | Adr  | reCmd<br>powiedział: | Stan | Dane[]                              | Przełącznik<br>CRC-16 |     |
|--------|------|----------------------|------|-------------------------------------|-----------------------|-----|
|        |      |                      |      | CFGData<br>(w języku<br>angielskim) |                       |     |
| 0x05+n | 0xXX | 0xEB                 | 0x00 | n(bajt)                             | LSB                   | MSB |

**CFGData:** Zawartość parametru, zobacz **Konfiguracja parametrów czytnika**, aby uzyskać szczegółowe informacje.

Rozpocznij szybką inwentaryzację

Komputer hosta służy do uruchamiania szybkiej inwentaryzacji, obecnie obsługuje tylko **ramkę poleceń serii Ex10**:

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
|      |      |      | cel    |                       |     |
| 0x05 | 0xXX | 0x50 | 0xXX   | LSB                   | MSB |

**Cel:** 1 bajt, 0 – cel A; 1-tarcza B;

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x50                 | 0x00 | --     | LSB                   | MSB |

Jeśli zapytany zostanie o tag, który pasuje do wstępnie zdefiniowanego warunku, czytnik automatycznie prześle dane w następującym formacie:

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0xXX | 0xXX | 0xee                 | 0x00 | --     | LSB                   | MSB |

parametry Danych[] są pokazane jak poniżej,

|        |
|--------|
| Dane[] |
|--------|

| Mrówka | Len  | Świadectwo<br>EPC/TID | Wskaźnik<br>RSSI<br>(RSSI) |
|--------|------|-----------------------|----------------------------|
| 0xXX   | 0xXX | N bajtów              | 0xXX                       |

**Mrówka:** wskazuje, która antena wysłała zapytanie o tag. Na przykład 0x05 to 0101 w systemie binarnym, co oznacza Antenę 3 i Antenę 1.

**Len:** 1 bajt, długość bajtu EPC/TID.

**EPC / TID:** numer EPC / dane TID tagu, długość danych powinna być identyczna z Len. Jednostką danych EPC C1 G2 jest słowo. Każdy numer EPC tagu lub dane TID jest najbardziej znaczącym słowem i najbardziej znaczącym bajtem w słowie jako pierwsze.

**RSSI:** 1 bajt, siła sygnału czytnika w momencie zapytania o tag.

Zatrzymaj szybką inwentaryzację

Komputer hosta służy do zatrzymywania szybkiej inwentaryzacji, obecnie obsługuje tylko serię Ex10. **Ramka poleceń:**

| Len  | Adr  | Cmd  | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|------|--------|-----------------------|-----|
| 0x04 | 0xXX | 0x51 | --     | LSB                   | MSB |

**Ramka odpowiedzi:**

| Len  | Adr  | reCmd<br>powiedział: | Stan | Dane[] | Przełącznik<br>CRC-16 |     |
|------|------|----------------------|------|--------|-----------------------|-----|
| 0x05 | 0xXX | 0x51                 | 0x00 | --     | LSB                   | MSB |

Indywidualne polecenia czytnika

Wszystkie parametry konfiguracyjne czytnika są w nim wymienione, co jest wygodne dla użytkowników do ustawiania i odczytywania w ujednolicony sposób.

Aby odczytać parametry, użyj polecenia get reader parameter i użyj polecenia set reader configuration w celu ustawienia parametrów. Po zmodyfikowaniu konfiguracji, z wyjątkiem konfiguracji sieci, która musi zostać ponownie uruchomiona, aby została zastosowana, inne konfiguracje zaczynają obowiązywać natychmiast. (Obecnie obsługuje tylko serię Ex10)

Tabela parametrów konfiguracyjnych czytnika

Tabela konfiguracyjna:

| Nie. | Nazwa                                   | Długość(bajt) | opisać   |
|------|---|---------------|--|
| 7    | Parametr skanowania                     | 3             | Informacje o parametrach używanych w procesie skanowania |
| 8    | TagSkupienie                            | 1             | Przełącznik TagFocus                                     |
| 9    | QAndSession<br>(Sesja kontroli jakości) | 2             | Parametr Q i sesji                                       |
| 10   | PARAMETR PRYWATNOŚCI                    | 2             | Parametr TID   |
| 11   | Parametr maski                          | N             | Parametr maski   |

Opis funkcji parametru czytnika

### CFG7: Parametr skanowania

Ta konfiguracja służy do ustawiania parametrów używanych przez czytnik podczas procesu skanowania

| Bajt  | 0        | 1                | 2      |
|-------|----------|------------------|--------|
| Treść | Interwał | Czas przebywania | Hrabia |

|          |   |   |   |
|----------|---|---|---|
| Domyślny | 3 | 0 | 5 |
|----------|---|---|---|

**Interwał:** Czas interwału między zapytaniami, zakres 0 ~ 6, jednostka 10 ms;

**Czas przebywania:** czas zapytania, zakres wynosi 2 ~ 255, jednostka to 100 ms; gdy przełącznik tagfocus jest włączony, jest ustalony na 200 ms.

**Liczba:** liczba zapytań, ważna tylko dla S1, S2, S3, zakres wynosi 0 ~ 5;

### CFG8: Przełącznik TagFocus

Ta konfiguracja służy do ustawiania przełącznika TagFocus

| Bajt     | 0            |
|----------|--------------|
| Treść    | TagSkupienie |
| Domyślny | 0            |

**TagFocus:** 0-wyłącz, 1-włącz.

### CFG9: Q i sesja

Ta konfiguracja służy do ustawiania parametrów antykolizyjnych używanych przez czytnik podczas procesu szybkiego zapytania.

| Bajt     | 0 | 1     |
|----------|---|-------|
| Treść    | Q | Sesja |
| Domyślny | 6 | 1     |

**P:** zakres wynosi 0 ~ 15.

**Sesja:** 0-S0; 1-S1; 2-S2; 3-S3; 255-automatyczny;

### CFG10: Parametr TID

Ta konfiguracja służy do ustawiania parametrów TID używanych przez czytnik podczas procesu szybkiego zapytania.

| Bajt     | 0                     | 1      |
|----------|-----------------------|--------|
| Treść    | AdrTID<br>powiedział: | LenTID |
| Domyślny | 6                     | 1      |

**AdrTID:** Zapytaj o początkowy adres słowa obszaru TID.

**LenTID:** Zapytanie o liczbę słów danych w obszarze TID. LenTID mieści się w zakresie od 0 do 15. Gdy wartość wynosi 0, oznacza to zapytanie EPC.

### CFG11: Parametr maski

Ta konfiguracja służy do ustawiania parametru maski używanego przez czytnik podczas procesu szybkiego zapytania.

| Bajt     | MaskMem | MaskAdr | MaskLen | Dane maski |
|----------|---------|---------|---------|------------|
| Treść    | 1       | 2       | 1       | MaskLen    |
| Domyślny | 1       | 0x0020  | 0       | 0x00       |

**MaskMem:** 1 bajt, wskazanie obszaru maski.

0x01 – pamięć EPC;

0x02 – pamięć

TID; 0x03 –

Pamięć

użytkownika.

Wszystkie inne wartości są zarezerwowane, czytnik zwróci status błędu parametru w ramce odpowiedzi, jeśli w tym polu zostanie dostarczona inna wartość.

**MaskAdr:** 2 bajty, adres bitu wejściowego maski, prawidłowy zakres MaskAdr to 0 ~ 16383.

---

**MaskLen:** 1 bajt, długość bitowa maski (jednostka: bity).

**MaskData:** dane maski. Długość MaskData jest równa  $\text{MaskLen}/8$ . Jeśli MaskLen nie jest wielokrotnością liczby całkowitej 8, długość MaskData jest równa  $\text{int}[\text{MaskLen}/8]+1$ . Nieokreślone dolne cyfry znaczące należy wypełnić cyfrą 0.

