

Dokumentacja techniczna

Protokół komunikacji GRYF



GRYFSMART

wersja: 3.40
08.01.2026

Spis treści:

1 Wstęp	4
2 Ogólny opis protokołu komunikacji.....	4
2.1 Adresowanie modułów	4
2.2 Topologia i warstwa fizyczna.....	5
2.3 Propagacja ramek danych.....	5
2.4 Model pracy – komunikacja eventowa.....	6
2.5 Logika rozproszona	6
3 Komendy komend AT.....	7
3.1 Komendy konfiguracyjne	7
3.1.1 Komenda AT+key	7
3.1.2 Komenda AT+M	7
3.1.3 Komenda AT+InS	8
3.1.4 Komenda AT+InL	8
3.1.5 Komenda AT+InZb	8
3.1.6 Komenda AT+InLam	9
3.1.7 Komenda AT+Out.....	9
3.1.8 Komenda AT+OutL	10
3.1.9 Komenda AT+OutLID	10
3.1.10 Komenda AT+OutID.....	11
3.1.11 Komenda AT+OutRID	11
3.1.12 Komenda AT+T1	12
3.1.13 Komenda AT+T2	12
3.1.14 Komenda AT+Ret	13
3.1.15 Komenda AT+StepIn	13
3.1.16 Komenda AT+StepMax	14
3.1.17 Komenda AT+OutStep1.....	14
3.1.18 Komenda AT+OutStep2.....	14
3.1.19 Komenda AT+OutStep3.....	15
3.1.20 Komenda AT+OutStep4.....	15
3.1.21 Komenda AT+OutStep5.....	15
3.1.22 Komenda AT+Init	16
3.1.23 Komenda AT+Time	16

3.1.24 Komenda AT+Save	17
3.1.25 Komenda AT+ActivIn	17
3.1.26 Komenda AT+Search	17
3.1.27 Komenda AT+Find	18
4.2 Komendy statusowe	18
4.2.1 Komenda I	18
4.2.2 Komenda O	19
4.2.3 Komenda R	19
4.2.4 Komenda L	19
4.2.5 Komenda AT+Time	20
4.2.6 Komenda T	20
4.2.7 Komenda PING	21
4.2.8 Komenda version	21
4.3 Komendy sterujące	21
4.3.1 Komenda SetLamp	21
4.3.2 Komenda SetOut	22
4.3.3 Komenda SetLED	22
4.3.4 Komenda AT+SetTime	22
4.3.5 Komenda AT+SetRol	23
4.3.6 Komenda AT+SMT (SetMaskTime)	23
4.3.7 Komenda AT+RST	24
4.3.8 Komenda PING	24
5 Kontrola, walidacja rozkazów- ERROR	26
6 Wsparcie techniczne	28



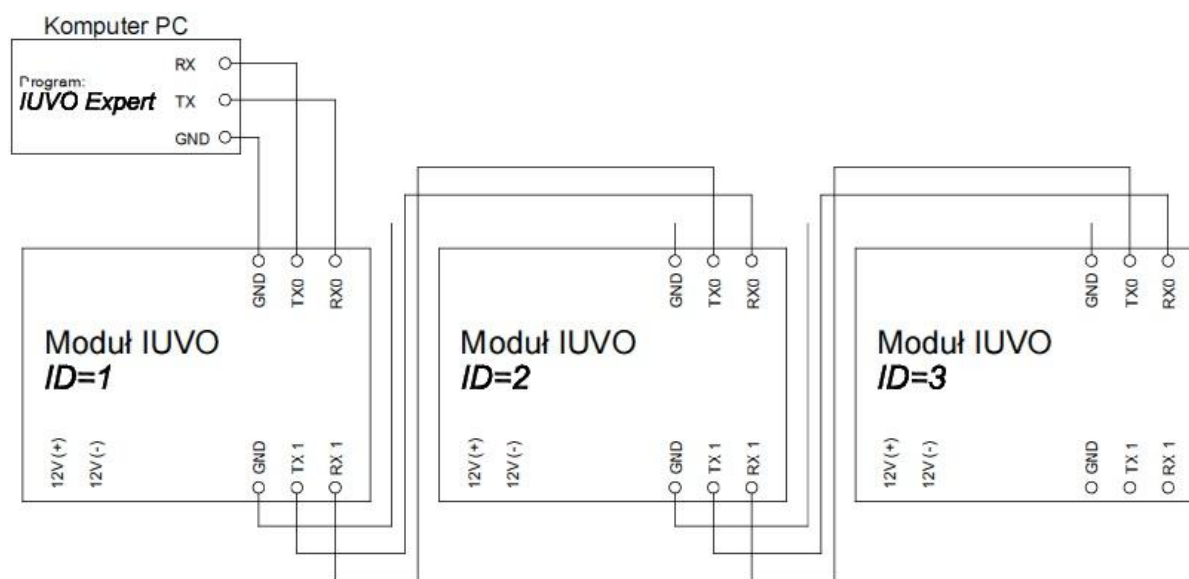
Wersja: 2.27
04.2026

1 Wstęp

Niniejszy dokument opisuje protokół komunikacji pomiędzy modułami systemu **GRYF** oraz pomiędzy modułem a komputerem klasy PC lub innym urządzeniem wyposażonym w interfejs RS232.

Każdy moduł wyposażony jest w dwa złącza komunikacyjne pracujące w standardzie elektrycznym **RS232 (± 12 V)**.

Sposób połączenia sterowników w sieć oraz ich podłączenia do komputera przedstawiono na rysunku 1.



Rys. 1. Sposób połączenia modułów GRYF w sieć komunikacyjną.

2 Ogólny opis protokołu komunikacji

Konfiguracja modułu realizowana jest poprzez wysyłanie tekstowych komend AT.

Każda komenda musi być zakończona znakiem ENTER:

CR = 13 (0x0D)

Urządzeniem wysyłającym komendy może być komputer klasy PC, sterownik nadrzędny lub inne urządzenie wyposażone w interfejs RS232.

Komunikacja odbywa się w trybie asynchronicznym.

Od strony elektrycznej wykorzystywane są trzy linie sygnałowe:

- **RX** – odbiór danych
- **TX** – transmisja danych
- **GND** – masa sygnałowa

2.1 Adresowanie modułów

Każdy moduł w sieci posiada unikalny adres identyfikacyjny **ID**, nadawany podczas pierwszej konfiguracji przy użyciu polecenia:

AT+Search

Numeracja rozpoczyna się od wartości:

ID = 1

dla pierwszego modułu w magistrali, a kolejne urządzenia otrzymują kolejne numery:

ID = 2, ID = 3, ..., ID = n

Adres ID umożliwia:

- jednoznaczny identyfikację urządzenia,
- kierowanie komend sterujących,
- adresowanie odpowiedzi,
- realizację komunikacji pomiędzy dowolnymi modułami w sieci.

System **GRYFSMART** stanowi **rozproszony system automatyki z lokalną logiką i komunikacją eventową**.

Każdy moduł pracuje autonomicznie, realizując przypisane funkcje logiczne oraz wymieniając informacje z innymi modułami za pośrednictwem magistrali komunikacyjnej.

2.2 Topologia i warstwa fizyczna

Magistrala zrealizowana jest w topologii liniowej (*daisy-chain*).

Każdy moduł posiada dwa porty komunikacyjne:

- **Port 0**
- **Port 1**

Moduły łączone są szeregowo:

ID1 ↔ ID2 ↔ ID3 ↔ ID4 ↔ ... ↔ IDn

Warstwa fizyczna oparta jest na interfejsie:

- **RS232 (±12 V)**
- transmisja asynchroniczna
- tryb pracy: **full-duplex**

Dzięki pełnemu duplexowi każdy moduł może jednocześnie nadawać i odbierać dane.

2.3 Propagacja ramek danych

Magistrala zrealizowana jest w topologii liniowej (*daisy-chain*).

Każdy moduł posiada dwa porty komunikacyjne:

Każdy moduł pełni podwójną rolę:

- urządzenia końcowego (node),
- transparentnego przekaźnika ramek (repeater).

Mechanizm działania:

1. Moduł odbiera ramkę na jednym z portów.
 2. Analizuje adres docelowy (ID).
 3. Jeżeli ID jest zgodne – przetwarza komendę lokalnie.
 4. Jeżeli ID jest różne – przekazuje ramkę dalej drugim portem.
- **Przykład komunikacji**

Transmisja z modułu ID4 do ID2:

ID4 → ID3 → ID2

Odpowiedź:



ID2 → ID3 → ID4

Ramki mogą propagować się w obu kierunkach magistrali.

2.4 Model pracy – komunikacja eventowa

System działa w oparciu o model zdarzeniowy (*event-driven*).

Transmisja inicjowana jest wyłącznie w wyniku wystąpienia zdarzenia – nie jest wymagane cykliczne odpytywanie urządzeń (brak pollingu).

Przykładowy przebieg:

1. Naruszenie wejścia w module.
2. Wykrycie zdarzenia przez moduł.
3. Uruchomienie przypisanej funkcji Fx.
4. Funkcja może:
 - sterować wyjściami lokalnymi,
 - wysłać komendę do innego modułu,
 - wygenerować komunikat statusowy.

Komunikaty statusowe mogą być generowane:

- synchronicznie (odpowiedź na zapytanie),
- asynchronicznie (automatycznie po zmianie stanu).

2.5 Logika rozproszona

Każdy moduł systemu **GRYFSMART** posiada własny mechanizm funkcji logicznych.

Zależności przyczynowo-skutkowe pomiędzy wejściami, funkcjami i wyjściami definiowane są w oprogramowaniu **GRYFS Expert**.

Po zapisaniu konfiguracji:

- logika przechowywana jest w pamięci nieulotnej modułu,
- system może pracować autonomicznie,
- nie jest wymagany centralny sterownik.

Dzięki temu:

- awaria jednego modułu nie blokuje całego systemu,
- możliwa jest skalowalna rozbudowa sieci,
- reakcje realizowane są lokalnie, bez opóźnień centralnego przetwarzania.

3 Komendy komend AT.

Komendy zostały podzielone na trzy grupy:

- **Komendy konfiguracyjne**, są to komendy które ustawiają parametry modułu (parametry wejść, parametry funkcji itd.). Do komend sterowniczych zalicza się również komenda AT+Search oraz AT+Find.
- **Komendy statusowe**, są to komendy wysyłane przez moduły informujące o stanie modułu (aktywne wejście, wyjścia, stan lampek, wersja softu)
- **Komendy sterujące**, są to komendy używane do sterowania modułami (zapalenie, zgaszenie, mrugnięcie lampek itd.). Dzięki nim można integrować system IUVO z innymi systemami.

3.1 Komendy konfiguracyjne

3.1.1 Komenda AT+key

```
AT+key=ID ,Nr_key ,T1 ,T2 ,typ<CR>
```

opis:

*Komenda ustawia parametry wejścia.
Id=1..255 Id konfigurowanego modułu
Nr_key=1..8 numer wejścia, fizyczny numer przycisku
T1=1..200 czas krótkiego naciśnięcia*10ms
T2=1..250 czas długiego naciśnięcia*10ms
typ=0..1 0:typ NO, 1:Typ NC*

przykład:

```
AT+key=6,2,5,20,0<CR>  
// Konfiguracja modułu o ID=6.Ustawienie parametrów wejścia 2:  
czas krótkiego naciśnięcia 50ms, czas długiego naciśnięcia  
200ms, typ wejścia NO
```

3.1.2 Komenda AT+M

```
AT+M=ID ,Fx ,My<CR>
```

opis:

*Komenda ustawia tryb pracy funkcji Fx.
ID=1..255 ID konfigurowanego modułu
Fx=1..16 numer funkcji 1..16
My=1..8- numer trybu 1..7 (1: Włącz, 2:Wyłącz, 3:Przełącz, 4:Włącz na czas,
5: Włącz na T1, wyłącz po T2, 6:*

przykład:

```
AT+M=1,2,5<CR>
```



//Konfiguracja modułu o ID=1.Ustawienie trybu pracy funkcji 2 na 5

3.1.3 Komenda AT+InS

AT+InS=ID ,Fx ,y1 ,y2 ,y3 ,y4 ,y5 ,y6 ,y7 ,y8<CR>

opis:

Komenda ustawia, czy funkcja Fx, będzie aktywowana, przez krótkie naciśnięcie danego wejścia.

ID=1..255 ID konfigurowanego modułu

Fx=1..16 oznacza numer funkcji 1..16

y1=0..1 (Aktywne, Nieaktywne) Wejście 1

y2=0..1 (Aktywne, Nieaktywne) Wejście 2

:

y8=0..1 (Aktywne Nieaktywne) Wejście 8

przykład:

```
AT+InS=3,10,0,1,1,0,0,0,0,1<CR>
```

```
//Ustawienia dla modułu o ID=3. Aktywacja funkcji 10,  
poprzez krótkie naciśnięcie wejścia 2,3 oraz 8
```

3.1.4 Komenda AT+InL

AT+InL=ID ,Fx ,y1 ,y2 ,y3 ,y4 ,y5 ,y6 ,y7 ,y8<CR>

opis:

Komenda ustawia czy funkcja Fx, będzie aktywowana przez długie przytrzymanie danego wejście.

ID=1..255 ID konfigurowanego modułu

x=1..16 oznacza numer funkcji

y1=0..1 (1: Aktywne,0: Nieaktywne) Wejście 1

y2=0..1 (1: Aktywne,0: Nieaktywne) Wejście 2

:

y8=0..1(1: Aktywne,0: Nieaktywne) Wejście 8

przykład:

```
AT+InL=4,3,0,1,1,0,0,0,0,1<CR>
```

```
//Ustawienia dla modułu o ID=4. Aktywacja funkcji 3,  
poprzez długie naciśnięcie wejścia 2,3 oraz 8
```

3.1.5 Komenda AT+InZb

AT+InZb=ID ,Fx ,y1 ,y2 ,y3 ,y4 ,y5 ,y6 ,y7 ,y8<CR>

opis:

Komenda określa, czy funkcja Fx zostanie aktywowana w wyniku zamknięcia (zwarcia) czy otwarcia (rozwarcia) styku bezpotencjałowego przypisanego do danego wejścia.

ID=1..255 ID konfigurowanego modułu

x=1..16 oznacza numer funkcji
y1=0..1,2 (0: brak akcji, 1: aktywacja „zwarciem”, 2: aktywacja „rozwarciem”) Wejście 1
y2=0..1,2 (0: brak akcji, 1: aktywacja „zwarciem”, 2: aktywacja „rozwarciem”) Wejście 2
:
y8=0..1 (0: brak akcji, 1: aktywacja „zwarciem”, 2: aktywacja „rozwarciem”) Wejście 8

przykład:

```
AT+InZb=4,3,0,1,1,0,0,0,2,1<CR>
//Ustawienia dla modułu o ID=4. Aktywacja funkcji 3,
poprzez długie „zwarciem” wejścia 2,3 oraz 8 oraz rozwarciem
wejścia 7
```

3.1.6 Komenda AT+InLam

AT+InLam=ID, Fx, y1, y2, y3, y4, y5, y6, y7, y8<CR>

opis:

Komenda ustawia czy funkcja Fx, będzie aktywowana, przez zapalenie lampki.

ID=1..255 ID konfigurowanego modułu

x=1..16 oznacza numer funkcji

y1=0..1 (Aktywne, Nieaktywne) Lampka 1

y2=0..1 (Aktywne, Nieaktywne) Lampka 2

:

y8=0..1 (Aktywne, Nieaktywne) Lampka 8

przykład:

```
AT+InLam=2,5,0,0,1,0,0,0,1,0<CR>
//Ustawienia dla modułu o ID=2. Ustawienia dla modułu o
ID=2. Aktywacja funkcji 5, poprzez lampki:3,7
```

3.1.7 Komenda AT+Out

Moduł IUVO Controller 0806

AT+Out=ID, Fx, y1, y2, y3, y4, y5, y6<CR>

opis:

Komenda ustawia, które wyjścia będą sterowane przez funkcję Fx

ID=1..255 ID konfigurowanego modułu

Fx=1..16 oznacza numer funkcji

y1=0..1 (Aktywne, Nieaktywne) Wyjście 1

y2=0..1 (Aktywne, Nieaktywne) Wyjście 2

:

y6=0..1 (Aktywne, Nieaktywne) Wyjście 8

przykład:



```
AT+Out=2,7,1,1,1,0,0,0<CR>
```

```
////Ustawienia dla modułu o ID=2. Funkcja 7 steruje wyjściami 1,2 oraz 3
```

Moduł GRYF S 0804

```
AT+Out=ID, Fx, y1, y2, y3, y4, y5, y6<CR>
```

opis:

Komenda ustawia, które rolety będą sterowane przez funkcję Fx

ID=1..255 ID konfigurowanego modułu

x=1..16 oznacza numer funkcji

*y1=0..1 (Nie Sterowana, Sterowana) **Roleta 1***

*y2=0..1 (Nie Sterowana, Sterowana) **Roleta 2***

*y3=0..1 (Nie Sterowana, Sterowana) **Roleta 3***

*y4=0..1 (Nie Sterowana, Sterowana) **Roleta 4***

y5=0..1 Bit nie używany

y6=0..1 Bit nie używany

przykład:

```
AT+Out=2,5,1,1,0,1,0,0<CR>
```

```
////Ustawienia dla modułu o ID=2. Funkcja 5, steruje roletami 1,2 oraz 4
```

3.1.8 Komenda AT+OutL

```
AT+OutL=ID, Fx, y1, y2, y3, y4, y5, y6, y7, y8<CR>
```

opis:

Komenda ustawia, które lampki będą zapalane, gaszone przez funkcję Fx

ID=1..255 ID konfigurowanego modułu

x=1..16 oznacza numer funkcji

y1=0..3 / 0:brak reakcji; 1:Zapal; 2:Zgas; 3:Mrugnij -> Lampka 1,

y2=0..3

:

y8=0..3 / 0:brak reakcji; 1:Zapal; 2:Zgas; 3:Mrugnij ->Lampka 8

przykład:

```
AT+OutL=3,7,1,1,1,2,0,0,0,0<CR>
```

```
//Ustawienia dla modułu o ID=3. Funkcja 7: zapala lampki 1,2,3; gasi lampke 4; 5,6,7,8 nie zminia stanu
```

3.1.9 Komenda AT+OutLID

```
AT+OutLID=ID, ID_cel, Fx, y1, y2, y3, y4, y5, y6, y7, y8<CR>
```

opis:

Komenda (MASKA)ustala jakie lampki, zostaną zapalone/ zgaszone/ mrugniete w module o ID=ID_cel. Pomaga utworzyć komendę AT+SetLamp.

ID=1..255 ID konfigurowanego modułu

ID_cel=1..32 numer ID modułu w którym będzie aktywowane zdarzenie
Fx=1..16 numer funkcji 1..16, która będzie aktywowała zdarzenie
y1=0..3 / 0:brak reakcji; 1:Zapał; 2:Zgas; 3:Mrugnij -> Lampka 1,
y2=0..3

:
y8=0..3 / 0:brak reakcji; 1:Zapał; 2:Zgas; 3:Mrugnij ->Lampka 8

przykład:

```
AT+OutL=5,20,7,1,1,1,2,0,0,0,0<CR>
```

```
//Ustawienia dla modułu o ID=5. Funkcja 7: zapala lampki  
1,2,3; gasi lampke 4; 5,6,7,8 nie zmienia stanu, w module o ID  
20.
```

3.1.10 Komenda AT+OutID

AT+OutID=ID, ID_cel, Fx, y1, y2, y3, y4, y5, y6<CR>

opis:

Komenda (MASKA)ustala jakie wyjścia, będą sterowane (włącz, wyłącz, przełącz) w module o ID=ID_cel. Pomaga utworzyć komendę AT+SetOut
ID=1..255 ID konfigurowanego modułu

ID_cel=1..32 numer ID modułu w którym będzie aktywowane zdarzenie
Fx=1..16 numer funkcji 1..16, która będzie aktywowała zdarzenie
y1=0..3 / 0:brak reakcji; 1:Włącz; 2:Wyłącz; 3:Przełącz -> Wyjście 1,
y2=0..3

:
Y6=0.. 3 / 0:brak reakcji; 1:Włącz; 2:Wyłącz; 3:Przełącz -> Wyjście 6.

przykład:

```
AT+OutID=5,20,7,1,1,1,2,3,0<CR>
```

```
//Ustawienia dla modułu o ID=5. Funkcja 7: włączenia wyjścia  
1,2,3; zgaszenia wyjścia 4; przełączenia wyjścia 5, braku  
reakcji wyjścia 6 w module o ID 20.
```

3.1.11 Komenda AT+OutRID

AT+OutRID=ID, ID_cel, Fx, T_ruchu, r1, r2, r3, r4 <CR>

opis:

Komenda (MASKA)ustala jakie rolety, będą sterowane (otwórz, zamknij, stop, step) w module o ID=ID_cel.

Pomaga utworzyć komendę AT+SetRol

ID=1..255 ID konfigurowanego modułu

ID_cel=1..255 numer ID modułu w którym będzie aktywowane zdarzenie

Fx=1..16 numer funkcji 1..16, która będzie aktywowała zdarzenie

T_ruchu=1..3599 / Czas ruchu rolety w sekundach

r1=0..4 / 0:brak reakcji; 1:Zamknij, 2: Otwórz, 3: Stop, 4: STEP (Roleta1)

r2=0..4 / 0:brak reakcji; 1:Zamknij, 2: Otwórz, 3: Stop, 4: STEP (Roleta2)

r3=0..4 / 0:brak reakcji; 1:Zamknij, 2: Otwórz, 3: Stop, 4: STEP (Roleta3)

r4=0..4 / 0:brak reakcji; 1:Zamknij, 2: Otwórz, 3: Stop, 4: STEP (Roleta4)



przykład:

```
AT+OutRID=5,20,7,4,2,2,1<CR>
```

```
//Ustawienia dla modułu o ID=5 (Maski), Funkcja 7; uruchamia trybu STEP rolety 1, Otwieranie rolety 2 i 3, Zamykanie rolety 4, w module o ID 20.
```

3.1.12 Komenda AT+T1

Moduł GRYF R0806T

AT+T1=ID, Fx, g, m, s<CR>

opis:

Komenda ustawia, czas parametru T1, funkcji Fx.

ID=1..255 ID konfigurowanego modułu

x=1..16 oznacza numer funkcji

g=0..255 godzina (0-255 godzin)

m=0..59 minuta (0-59 minut)

s=0..59 sekunda (0-59 sekund)

przykład:

```
AT+T1=5,2,2,40,10<CR>
```

```
//Ustawienia dla modułu o ID=5. Ustawianie czasu T1 na wartość 2 godziny,40 minut, 10 sekund, dla funkcji 2
```

Moduł GRYF S 0804

AT+T1=ID, Fx, g, m, s<CR>

opis:

Komenda ustawia, czas (czas T1) po jakim ma się otworzyć roleta, funkcji Fx.

ID=1..255 ID konfigurowanego modułu

x=1..16 oznacza numer funkcji

g=0..255 godzina (0-255 godzin)

m=0..59 minuta (0-59 minut)

s=0..59 sekunda (0-59 sekund)

przykład:

```
AT+T1=7,2,2,40,10<CR>
```

```
//Ustawienia dla modułu o ID=7. Ustawianie czasu jaki ma upłynąć do rozpoczęcia ruchu rolety w funkcji 2.
```

3.1.13 Komenda AT+T2

Moduł GRYF R0806

AT+T2=ID, Fx, g, m, s<CR>

opis:

Komenda ustawia, czas parametru T2, funkcji Fx.

ID=1..255 ID konfigurowanego modułu

x=1..16 oznacza numer funkcji

g=0..255 godzina (0-255 godzin)
m=0..59 minuta (0-59 minut)
s=0..59 sekunda (0-59 sekund)

przykład:

```
AT+T2=2,3,0,20,10<CR>
```

```
//Ustawienia dla modułu o ID=2.Ustawianie czasu T2 na wartość  
0 godziny,20 minut, 10 sekund, dla funkcji 3
```

Moduł GRYF R 0804

AT+T2=ID , Fx , g , m , s<CR>

opis:

Komenda ustawia, czas ruchu rolety (czas T2), funkcji Fx.

ID=1..255 ID konfigurowanego modułu

x=1..16 oznacza numer funkcji

g =0..59 Bit nie używany

m=0..59 minuta (0-59 minut)

s=0..59 sekunda (0-59 sekund)

przykład:

```
AT+T2=2,3,0,1,10<CR>
```

```
//Ustawienia dla modułu o ID=2.Ustawianie czasu ruchu rolety  
T2 na wartość 1 minut, 10 sekund, dla funkcji 3
```

3.1.14 Komenda AT+Ret

AT+Ret=ID , Fx , y<CR>

opis:

Komenda ustawia, sposób odmierzenia czasu.

ID=1..255 ID konfigurowanego modułu

x=1..16 oznacza numer funkcji

y=1- kolejny impuls powoduje zliczanie czasu od nowa

y=0- kolejny impuls nie powoduje zliczania czasu od nowa

przykład:

```
AT+Ret=3,2,1<CR>
```

```
//Ustawienia dla modułu o ID=3.Funkcja 2, odmierza czas w  
sposób retrygowalny
```

3.1.15 Komenda AT+StepIn

AT+StepIn=ID , Sx , y1 , y2 , y3 , y4 , y5 , y6 , y5 , y6 , y7 , y8<CR>

opis:

Komenda ustawia, które wejścia będą aktywowały sekwencje.(maska)

ID=1..255 ID konfigurowanego modułu

Sx=1..8 oznacza numer Sekwencji

y1=0..1 / 0:brak aktywacji; 1:aktywacja

y2=0..1 / 0:brak aktywacji; 1:aktywacja



Y8=0..1 / 0:brak aktywacji; 1:aktywacja

przykład:

```
AT+ StepIn=3,7,1,0,1,1,0,0,0,0<CR>
//Ustawienia dla modułu o ID=3. Wejście 1,3,4 będzie
aktywowało sekwencję numer 7
```

3.1.16 Komenda AT+StepMax

AT+StepMax=ID , Fx , y<CR>

opis:

*Komenda ustawia, ilość kroków .
ID=1..255 ID konfigurowanego modułu
x=1..16 oznacza numer funkcji
y=1..5 ilość kroków*

przykład:

```
AT+StepMax=2,2,4<CR>
//Ustawienia dla modułu o ID=2. Sekwencja 2, posiada 4 kroki
step.
```

3.1.17 Komenda AT+OutStep1

AT+OutStep1=Fx , y1 , y2 , y3 , y4 , y5 , y6<CR>

opis:

*Komenda ustawia, które wyjścia będą aktywowane w kroku step1
ID=1..255 ID konfigurowanego modułu
x=1..8 oznacza numer Sekwencji
y1=0..2 / 0:brak reakcji; 1:Włącz; 2:Wyłącz -> Wyjście 1,
y2=0..2 / 0:brak reakcji; 1:Włącz; 2:Wyłącz -> Wyjście 2,
:
Y6=0..2 / 0:brak reakcji; 1:Włącz; 2:Wyłącz -> Wyjście 6*

przykład:

```
AT+OutStep1=2,7,1,1,1,2,0,0<CR>
//Ustawienia dla modułu o ID=2. Sekwencja 7 w kroku 1, włączy
wyjścia 1,2,3; wyłączy wyjście 4, stan pozostałych wyjść
zostanie niezmienny
```

3.1.18 Komenda AT+OutStep2

AT+OutStep2=ID , Fx , y1 , y2 , y3 , y4 , y5 , y6<CR>

opis:

*Komenda ustawia, które wyjścia będą aktywowane w kroku step2
ID=1..255 ID konfigurowanego modułu
x=1..8 oznacza numer Sekwencji
y1=0..2 / 0:brak reakcji; 1:Włącz; 2:Wyłącz -> Wyjście 1,
y2=0..2 / 0:brak reakcji; 1:Włącz; 2:Wyłącz -> Wyjście 2,*

:
Y6=0..2 / 0:brak reakcji; 1:Włącz; 2:Wyłącz -> Wyjście 6

przykład:

```
AT+OutStep2=3,7,0,1,1,0,0,0<CR>
```

//Ustawienia dla modułu o ID=3. Sekwencja 7 w kroku 2, włączy wyjścia 2,3; stan pozostałych wyjść zostanie niezmienny

3.1.19 Komenda AT+OutStep3

AT+OutStep3=ID, Fx, y1, y2, y3, y4, y5, y6<CR>

opis:

Komenda ustawia, które wyjścia będą aktywowane w kroku step3

ID=1..255 ID konfigurowanego modułu

x=1..8 oznacza numer Sekwencji

y1=0..2 / 0:brak reakcji; 1:Włącz; 2:Wyłącz -> Wyjście 1,

y2=0..2 / 0:brak reakcji; 1:Włącz; 2:Wyłącz -> Wyjście 2,

:

Y6=0..2 / 0:brak reakcji; 1:Włącz; 2:Wyłącz -> Wyjście 6

przykład:

```
AT+OutStep3=1,7,0,1,1,0,1,1<CR>
```

//Ustawienia dla modułu o ID=1. Sekwencja 7 w kroku 3, włączy wyjścia 2,3,5,6; stan pozostałych wyjść zostanie niezmienny

3.1.20 Komenda AT+OutStep4

AT+OutStep4=ID, Fx, y1, y2, y3, y4, y5, y6<CR>

opis:

Komenda ustawia, które wyjścia będą aktywowane w kroku step4

ID=1..255 ID konfigurowanego modułu

x=1..8 oznacza numer Sekwencji

y1=0..2 / 0:brak reakcji; 1:Włącz; 2:Wyłącz -> Wyjście 1,

y2=0..2 / 0:brak reakcji; 1:Włącz; 2:Wyłącz -> Wyjście 2,

:

Y6=0..2 / 0:brak reakcji; 1:Włącz; 2:Wyłącz -> Wyjście 6

przykład:

```
AT+OutStep4=6,7,1,1,1,1,1,1<CR>
```

//Ustawienia dla modułu o ID=6. Sekwencja 7 w kroku 4, włączy wyjścia 2,3,5,6; stan pozostałych wyjść zostanie niezmienny

3.1.21 Komenda AT+OutStep5

AT+OutStep5=ID, Fx, y1, y2, y3, y4, y5, y6<CR>

opis:

Komenda ustawia, które wyjścia będą aktywowane w kroku step4



ID=1..255 ID konfigurowanego modułu
x=1..8 oznacza numer Sekwencji
y1=0..2 / 0:brak reakcji; 1:Włącz; 2:Wyłącz -> Wyjście 1,
y2=0..2 / 0:brak reakcji; 1:Włącz; 2:Wyłącz -> Wyjście 2,
:
Y6=0..2 / 0:brak reakcji; 1:Włącz; 2:Wyłącz -> Wyjście 6

przykład:

```
AT+OutStep5=6,5,2,2,2,1,1,1<CR>
//Ustawienia dla modułu o ID=6. Sekwencja 5 w kroku 5,
wyłączyć wyjścia 1,2,3 oraz włączyć wyjścia 4,5,6
```

3.1.22 Komenda AT+Init

AT+Init=ID, Out1, Out2, Out3, Out4, Out5, Out6<CR>

opis:

Komenda definiuje sposób inicjalizacji wyjść modułu po restarcie lub podaniu napięcia zasilania, określając czy wyjścia zostaną ustawione w stan aktywny (ON), nieaktywny (OFF)

ID=1..255 ID konfigurowanego modułu
y1=0..1 / 0:Wyłącz; 1:Włącz (IUVO Controller)
/ 0: Zamknij; 1:Otwórz; 2: Brak reakcji (IUVO RollerShutter)

przykład:

```
//Komenda wysłana do modułu GRYF R0806
//AT+Init=2,1,0,0,0,1,1<CR>
//Ustawienia dla modułu o ID=2. Po powrocie zasilania
włączenie wyjścia 1,5,6 oraz wyłączenie wyjścia 2,3,4
//Komenda wysłana do modułu GRYF S0804
//AT+Init=5,2,1,0,1,0,0<CR>
//Ustawienia dla modułu o ID=5. Po powrocie zasilania roleta 3
//otwiera się, roleta 1 nie zmieni stanu, roleta 3 zamykają
się
//Dwa ostatnie pola (Out5, Out6) nie mają znaczenia, nie
dotyczą //rozkazu wysyłanego do modułu IUVO RollerShutter
```

3.1.23 Komenda AT+Time

AT+MTime= ID_modulu, Fx, g, m, s, nd<CR>

opis:

Ustawienie czasu,
ID_modulu=1..32 / informuje do którego modułu jest wysłane zapytanie
Fx=1..16 oznacza numer funkcji 1..16
g=0..23 / ..godzina
m=0..59 / ..minuty
s=0..59 / ..sekundy
nd=1..7 / nazwa dnia 1:poniedziałek, 2:wtorek...
:

```
AT+SetTime= 3,1,11,23,55,2<CR>
```

//Ustawienia dla modułu o ID=3, funkcji 1. Funkcja 1 zostanie wyzwolona we wtorek o godzinie 11:23:55

Uwagi:

Komenda dotyczy jedynie modułów posiadających zegar czasu rzeczywistego.

3.1.24 Komenda AT+Save

AT+Save=ID, I1_Mod<CR>

opis:

Komenda zapisuje ustawienia w pamięci EEPROM w module o ID(1..255)

I1_Mod - ilość modułów w Sieci

przykład:

AT+Save=2, 5<CR>

//Ustawienia dla modułu o ID=2. Komenda zapisuje ustawienia w pamięci EEPROM, w sieci jest 5 modułów.

3.1.25 Komenda AT+ActivIn

AT+ActivIn=ID, Fx, x1, x2, x3, x4, x5, x6<CR>

opis:

Komenda ustawia, jakie akcje, będą aktywowały wejścia (krótkie naciśnięcie, długie naciśnięcie, lampki)

ID=1..255 ID konfigurowanego modułu

x=1..16 oznacza numer funkcji

x1=0..1 Aktywacja wejścia za pomocą krótkiego naciśnięcia

x2=0..1 Aktywacja wejścia za pomocą długiego naciśnięcia

x3=0..1 Aktywacja wejścia za pomocą Lampki

x4=0..1 Aktywacja wejścia poprzez „zwarcie”

x5=0..1 Aktywacja wejścia poprzez „rozwarcie”

x6=0..1 Aktywacja wejścia poprzez Timer

przykład:

AT+ActivIn=2, 3, 1, 1, 0, 0, 0, 0<CR>

//Ustawienia dla modułu o ID=2. Aktywacja funkcji 3, odbywa się przez, krótkie naciśnięcie oraz długie naciśnięcie

UWAGA:

Konfiguracja maski x6, dotyczy tylko modułów wyposażonych w Timer.

3.1.26 Komenda AT+Search

AT+Search=licznik, ID_modulu<CR>

opis:

Komenda jest wysyłana do modułu w celu nadania adresu ID modułowi oraz zapisanie go w pamięci EEPROM.

AT+Search=licznik, ID_modulu<CR>

licznik=1..32 / Zmienna inkrementowana w każdym module



ID_modulu=1..32 / informuje do którego modułu jest wysłane zapytanie

:

AT+Search= 1,3<CR>

//Komenda wysłana do modułu 3. Po przejściu przez moduł 1 zmienna licznik jest inkrementowana..

Przykład:

Wysłanie komendy: **AT+Search=0,5<CR>** jeżeli w sieci jest moduł o ID=5 to otrzymamy odpowiedz: **AT+Find=5,0605,3<CR>**.

3.1.27 Komenda AT+Find

AT+Find= ID_modulu,Soft,Typ<CR>

opis:

Odpowiedz na komendę AT+Search,

ID_modulu=1..32 / informuje do którego modułu jest wysłane zapytanie

Soft=0000...9999/ numer wsadu

Typ=1.9/

1: ...

2: ...

3: GRYF S0804

4: GRYF R0806T

5: GRYF R0808

:

AT+Find= 3,00F1,1<CR>

//Odpowiedz modułu, (typ: GRYF R0806T) na komendę AT+Search. ID modułu 3/SOFT=0821

Uwagi:

Jeżeli komenda AT+Search dotrze do poszukiwanego modułu, to moduł odsyła komendę AT+Find

Wówczas komenda trafia na UART1, moduł GRYF przesyła ją dalej na UART0 itd. Aż komenda trafi do PC i moduł się zarejestruje.

4.2 Komendy statusowe

Komendy statusowe realizują funkcję odczytu bieżącego stanu pracy modułu, zwracając informacje o stanie logicznym wejść, wyjść wykonawczych oraz stanie lampek.

W przypadku zmiany stanu wejścia, wyjścia modułu, komunikat statusowy generowany jest asynchronicznie jako odpowiedź na wykryte zdarzenie.

4.2.1 Komenda I

I=ID, In1, In2, In3, In4, In5, In6, In7, In8<CR>

opis:

Komenda statusowa informująca o stanie wejść w module.

ID=1..255 ID modułu z którego pobieramy informacje

In[x] Stan wejścia x[x=1..8]

1: wejście aktywne (wykryto zwarcie styku)
0: wejście nieaktywne (wykryto rozwarcie styku)

przykład:

```
I=2,1,1,1,0,0,0,0,0<CR>  
//W module o ID=2, wejścia 1,2,3 jest zwarte, pozostałe  
wejścia są rozwarte
```

UWAGA:

Komenda jest wysyłana w sposób cykliczny lub asynchronicznie pod wpływem zmiany stanu Wejścia

4.2.2 Komenda O

O=ID , O1 , O2 , O3 , O4 , O5 , O6<CR>

opis:

Komenda statusowa informująca o stanie wyjścia w module.

ID=1..255 ID modułu z którego pobieramy informacje

O[x] Stan wyjścia $x[x=1..6]$

1 – wyjście załączone (przełącznik wystawiony, styki zwarte)

0 – wyjście wyłączone (przełącznik niewystawiony, styki rozwarte)

przykład:

```
O=5,1,1,1,0,0<CR>  
//W module o ID=5, wyjścia 1,2,3 jest aktywne, pozostałe  
wyjścia są nieaktywne
```

UWAGA:

Komenda jest wysyłana w sposób cykliczny lub asynchronicznie pod wpływem zmiany stanu Wyjścia

4.2.3 Komenda R

R=ID , R1 , R2 , R3 , R4<CR>

opis:

Komenda statusowa informująca o stanie rolet w module.

ID=1..255 ID modułu z którego pobieramy informacje

R[x] Stan wyjścia $x[x=1..4]$

2: zamykanie rolety

1: otwieranie rolety

0: zatrzymana

przykład:

```
O=4,1,2,2,0<CR>  
//W module o ID=4, rolety 2,3 są zamykane, roleta 1 jest  
otwierana, roleta 4 jest zatrzymana
```

UWAGA:

Komenda jest wysyłana w sposób cykliczny lub asynchronicznie pod wpływem zmiany stanu Rolety

4.2.4 Komenda L

L=ID , L1 , L2 , L3 , L4 , L5 , L6 , L7 , L8<CR>



opis:

Komenda statusowa informująca o stanie zmiennych wirtualnych lampek w module.

ID=1..255 ID modułu z którego pobieramy informacje

L[x] Stan lampki $x[x=1..8]$

1: Lampka zapalona

0: Lampka zgaszona

przykład:

```
O=4,1,0,0,0,1,1,1,1<CR>
```

```
//W module o ID=4, lampki 1,5,6,7,8 są zapalone, pozostałe lampki są zgaszone
```

UWAGA:

Komenda jest wysyłana w sposób cykliczny lub asynchronicznie pod wpływem zmiany stanu Lampek.

4.2.5 Komenda AT+Time

```
AT+Time=g,m,s,NumerDnia<CR>
```

opis:

Informacja o godzinie

g=0..23 / ..godzina

m=0..59 / ..minuty

s=0..59 / ..sekundy

NumerDnia=1..7 / nazwa dnia 1:poniedziałek, 2:wtorek...

:

```
AT+SetTime= 3,14,2,3,11,23,55<CR>
```

```
//Ustawienie daty w module o ID=3, 2014-luty-2, 11:23:55
```

Uwagi:

Komenda jest wysyłana co 1 sekundę z modułu posiadającego zegar czasu rzeczywistego. Jest wysyłana na port0 oraz na port1. Moduły, które otrzymały komendę „wyluskują” informację (aktualna godz, min, sek, dzien) oraz przesyłają ją dale do następnych modułów

4.2.6 Komenda T

```
T=ID,Nr_czujnika,Czesc_calkowita,Czesc_dzies<CR>
```

opis:

Informacja o temperaturze z danego modułu

ID=1..255 ID modułu z którego pobieramy informacje

Nr_czujnika – numer czujnika temperatury

Czesc_calkowita – Wartość całkowita temperatury

Czesc_dzies – Wartość dziesiętna temperatury

```
T=3,1,21,3<CR>
```

```
//Czujnik 1 podłączony do modułu 3 wskazuje temperaturę 21,3C
```

Uwagi:

Moduł wysyła komendę „T” co (około) 5 sekund.

4.2.7 Komenda PING

PING=ID<CR>

opis:

*Sprawdzanie komunikacji z modulem
ID=1..255 ID modułu który jest testowany*

```
PING=5<CR>
```

```
//Moduł o ID=5 jest pingowany, jeżeli komunikacja jest poprawna moduł odpowie komendą PONG=5
```

4.2.8 Komenda version

version=ID, typ_modulu, soft_ver, data, czas<CR>

opis:

*Informacja o wersji softu w module o ID
ID=1..255 ID modułu z którego pobieramy informacje
soft_ver – wersja oprogramowania modułu
typ_modulu – 1,2,3,4
data – data generacji wsadu
czas – czas generacji wsadu*

```
version=id<CR>
```

```
//po wysłaniu zapytania version=id, moduł zwraca informacje  
version=id,typ_mod,soft_ver,data,czas
```

4.3 Komendy sterujące

4.3.1 Komenda SetLamp

AT+SetLamp=ID_cel, y1, y2, y3, y4, y5, y6, y7, y8<CR>

opis:

*Komenda steruje lampkami (Zapal, Zgaś, Mrugnij)
ID_cel=0..255 / 0: Wyślij do wszystkich modułów, 1..255 do modułu o ID=1..255
y1=0..3 / 0:brak reakcji; 1:Zapal; 2:Zgas; 3:Mrugnij -> Lampka 1,
y2=0..3
:
y8=0..3 / 0:brak reakcji; 1:Zapal; 2:Zgas; 3:Mrugnij ->Lampka 8*

przykład:

```
AT+SetLamp=7, 1, 1, 1, 2, 3, 0, 0, 0<CR>
```

```
//Wysterowanie modułu o ID=7; Zapalenie lampek1,2,3; Zgaszenie lampki 4; Mrugnięcie lampką 5; Brak reakcji lampka 6,7,8
```



4.3.2 Komenda SetOut

```
AT+SetOut=ID_cel,y1,y2,y3,y4,y5,y6<CR>
```

opis:

Komenda steruje wyjściami (Włącz, Wyłącz, Przełącz) w module GRYF R0806
ID_cel=0..255 / 0: Wyślij do wszystkich modułów, 1..255 do modułu o ID=1..255
y1=0..2 / 0: brak reakcji; 1: Włącz; 2: Wyłącz; 3: Przełącz – wyjście 1
y2=0..2
:
y6=0..2 / 0: brak reakcji; 1: Włącz; 2: Wyłącz; 3: Przełącz – wyjście 6

przykład:

```
AT+SetOuts= 3,1,2,1,2,0,0<CR>  
//Wysterowanie modułu o ID=3; Włączenie wyjścia 1,3;  
Wyłączenie wyjścia 2; Brak reakcji wyjść 5,6
```

4.3.3 Komenda SetLED

```
SetLED=ID_cel,kanał,poziom_jasności,stan_PWM<CR>
```

opis:

Komenda steruje wyjściem PWM w module GRYF R0806
ID_cel=1..255 / 0: Wyślij do modułu o ID=1..255
kanał=1,2 / 1: dla GRYF R0806T
poziom_jasności=0..100 / Zakres regulacji jasności (PWM): 0–100 % :
stan_PWM=1 / Parametr opcjonalny, jeżeli wartość jego jest 1, to otrzymamy
odpowiedź LED=ID,kanał,poziom_jasności

przykład 1:

```
SetLED= 4,1,90<CR>  
// Ustawienie wysterowania kanału 1 PWM na 90 % w module o ID=4
```

przykład 2:

```
SetLED= 4,1,90,1<CR>  
// Ustawienie wysterowania kanału 1 PWM na 90 % w module o  
ID=4, moduł o ID=4 odpowiada komendą LED=4,1,90
```

4.3.4 Komenda AT+SetTime

```
AT+SetTime= ID_modulu,Fx,g,m,s,nr_dnia<CR>
```

opis:

Komenda steruje lampkami (Zapal, Zgaś, Mrugnij)
ID_cel=1..255 / Wyślij do modułu o ID=1..255
y1=0..2 / 0: brak reakcji; 1: Włącz; 2: Wyłącz ->Wyjście 1; 3: Przełącz ->Wyjście 1
y2=0..2
:
y6=0..2 / 0: brak reakcji; 1: Włącz; 2: Wyłącz ->Wyjście 6; 3: Przełącz ->Wyjście 6

przykład:

```
AT+SetOuts= 3,1,2,1,2,0,0<CR>
```

```
//Wysterowanie modułu o ID=3; Włączenie wyjścia 1,3;  
Wyłączenie wyjścia 2; Brak reakcji wyjść lampka 5,6
```

Uwagi:

Komenda tyczy się jedynie modułów posiadających zegar czasu rzeczywistego. Ustawia godzinę i czas w module

4.3.5 Komenda AT+SetRol

```
AT+SetRol=ID_cel ,T_ruchu ,R1 ,R2 ,R3 ,R4 , SumKon<CR>
```

opis:

Komenda steruje roletami (Zamknij, Otwórz, Stop, STEP)

ID_cel=0..255 / 0: Wyślij do wszystkich modułów, 1..255 do modułu o ID=1..255

T_ruch=1..3599 / Czas ruchu rolety, podawany w sekundach

R1=0..4 / 0: brak reakcji; 1: Zamknij; 2: Otwórz; 3: Stop; 4: STEP (Roleta 1)

R2=0..4 / 0: brak reakcji; 1: Zamknij; 2: Otwórz; 3: Stop; 4: STEP (Roleta 2)

R3=0..4 / 0: brak reakcji; 1: Zamknij; 2: Otwórz; 3: Stop; 4: STEP (Roleta 3)

R4=0..4 / 0: brak reakcji; 1: Zamknij; 2: Otwórz; 3: Stop; 4: STEP (Roleta 4)

SumKon- Suma kontrolna służy do kontroli poprawności ramki. Kontroluje, czy żadna ze zmiennych nie uległa uszkodzeniu, zmianie.

Suma kontrolna= ID_cel+T_ruch+R1+R2+R3+R4

```
AT+SetRol=8,45,1,1,2,3,60<CR>
```

```
//W module o ID=8, Zamknij rolete 1,2; Otwórz rolete 3,  
Uruchom tryb STEP dla rolety 4, suma kontrolna=8+45+1+1+2+3=60
```

4.3.6 Komenda AT+SMT (SetMaskTime)

```
AT+SMT=
```

```
ID_modulu ,Fx ,d1 ,d2 ,d3 ,d4 ,d5 ,d6 ,d7 ,g ,m ,s<CR>
```

opis:

Ustawienie Maski czasu,

ID=1..255 ID konfigurowanego modułu

x=1..16 oznacza numer funkcji

d1...d7 – dzień tygodnia d1-poniedziałek, d2- wtorek... [0..1]

g=0..23 / ..godzina

m=0..59 / ..minuty

s=0..59 / ..sekundy

:

```
AT+SetTime= 3,14,1,1,1,1,0,0,0,11,25,21<CR>
```

```
//Ustawienie daty w module o ID=3, Aktywacja Funkcji 14, w  
poniedziałek, wtorek, środę i czwartek o godzinie 11:25:21.
```

Uwagi:

Komenda jest Maską informującą moduł, która funkcja o której godzinie ma być aktywowana oraz w jaki dzień.

Pierwsze bit w bajcie MaskTimeDay zawsze musi mieć wartość zero.

4.3.7 Komenda AT+RST

Komenda realizuje programowy restart modułu

AT+RST=ID_modulu<CR>

opis:

*Wysłanie komendy AT+RST=ID_Modulu, spowoduje restart modułu o ID=ID_Modulu
ID=1..255 ID Resetowanego modułu
ID=0 Restart wszystkich modułów w sieci*

Przykład 1:

```
AT+RST=5<CR>  
//Restart modułu o ID=5
```

Przykład 2:

```
AT+RST=0<CR>  
//Restart wszystkich modułów w sieci
```

Informacja o restarcie modułu (wysłanie komendy z modułu)

AT+RST=TypRestartu, ID_modulu<CR>

opis:

Moduł wyśle komendę AT+RST, jeżeli zostanie zrestartowany w sposób fizyczny poprzez zdjęcie napięcia lub w sposób programowy poprzez komendę AT+RST=ID_Modulu.

TypRestartu=

0: Fizyczny restart modułu

1: Restart modułu z komendy AT+RST

ID=1..255 ID Resetowanego modułu

Przykład 1- Fizyczny restart modułu:

```
AT+RST=0,5<CR>  
//Moduł o ID=5, został zrestartowany poprzez odłączenie zasilania
```

Przykład 2- Restart modułu za pomocą komendy AT+RST:

```
AT+RST=1,5<CR> //informacja o źródle restartu  
...po 2 sekundach  
AT+RST=0,5<CR> //informacja, że moduł został uruchomiony
```

4.3.8 Komenda PING

Jest to komenda sprawdzenia komunikacji z modulem. Po wysłaniu komendy PING, do modułu, moduł odpowiada powinien odpowiedzieć taką samą komendą, jeżeli komunikacja jest prawidłowa.

„Ping- gowanie” modułu (wysłanie komendy do modułu)

PING=ID_modulu<CR>

opis:

*Wysłanie komendy PING=ID_Modulu, spowoduje odesłanie komendy
PING=ID_Modulu*

ID=1..255 ID pingowanego modułu

Przykład 1:

PING=5<CR>

//Wysłanie komendy PING do modułu o ID=5, jeżeli komunikacja jest poprawna moduł odpowiada komendą PING=5

5 Kontrola, walidacja rozkazów- ERROR

Jeżeli komenda nie posiada właściwego formatu, to zostanie zwrócony komunikat o błędzie. Błąd może dotyczyć długości ramki, typu argumentów lub zakresu argumentów. Jeżeli komenda jest nieprawidłowa, nie zostanie wykonana.

ERROR=ID_modułu,ERROR_Ramki,ERROR_Nr<CR>

opis:

ID_modułu- Numer ID modułu w którym wykryto nieprawidłową komendę

ERROR_Ramki- Typ rozkazu w jakim wystąpił błąd

ERROR_Nr- Dokładny typ błędu.

- 1 – zła długość ramki
- 2 – ID modułu nie jest liczbą
- 3 – ID jest spoza zakresu 0..30
- 4 – parametr spoza zakresu
- 5 – parametr nie jest liczbą

Przykład:

```
>> AT+SetOut=1,1,2,3,3,3,4<CR>
```

```
<< ERROR=1,1,4<CR>
```

//Komunikat o błędzie został wysłany z modułu o ID=1, Błąd dotyczył błędnej ramki AT+SetOut, Błędny zakres parametru (błąd dotyczy wyjścia 6, parametr ma większą wartość. Jest spoza zakresu <0..3>).

Tabela błędów

Error Ramki	
ERROR=ID, 0, x<CR> x=1..5 Błąd ramki AT+SetOut	ERROR=id, 0, 1<CR> Błędna długość ramki ERROR ERROR=id, 0, 2<CR> ID nie jest liczbą ERROR=id, 0, 3<CR> Zły adres ID ERROR=id, 0, 4<CR> parametr error spoza zakresu ERROR=id, 0, 5<CR> parametr error nie jest liczbą
ERROR=ID, 1, x<CR> x=1..5 Błąd ramki AT+SetOut	ERROR=id, 1, 1<CR> Błędna długość ramki AT+SetOut ERROR=id, 1, 2<CR> ID nie jest liczbą ERROR=id, 1, 3<CR> Zły adres ID ERROR=id, 1, 4<CR> parametr out(1..6) spoza zakresu ERROR=id, 1, 5<CR> parametr out(1..6) nie jest liczbą
ERROR=ID, 2, x<CR> x=1..5 Błąd ramki AT+SetRol	ERROR=id, 2, 1<CR> Błędna długość ramki AT+SetRol ERROR=id, 2, 2<CR> ID nie jest liczbą ERROR=id, 2, 3<CR> Zły adres ID ERROR=id, 2, 4<CR> parametr rol(1..4) spoza zakresu ERROR=id, 2, 5<CR> parametr rol(1..4) nie jest liczbą
ERROR=ID, 3, x<CR> x=1..5 Błąd ramki AT+SetLamp	ERROR=id, 3, 1<CR> Błędna długość ramki AT+SetLamp ERROR=id, 3, 2<CR> ID nie jest liczbą ERROR=id, 3, 3<CR> Zły adres ID ERROR=id, 3, 4<CR> parametr Lamp(1..8) spoza zakresu ERROR=id, 3, 5<CR> parametr Lamp(1..8) nie jest liczbą



6 Wsparcie techniczne

Dokumentacja techniczna, aktualne wersje instrukcji oraz przykłady zastosowań modułów **GRYF** dostępne są na stronie internetowej:

www.gryfsmart.pl

W przypadku pytań technicznych lub potrzeby uzyskania dodatkowych informacji prosimy o kontakt:

E-mail: kontakt@gryfsmart.pl

Tel. kom.: +48 510 30 99 70