

Dokumentacja techniczna

Protokół komunikacji IUVO



wersja: 3.11
10.2021

Spis treści:

1 Wstęp	4
2 Ogólny opis protokołu komunikacji.....	4
4 Komendy komend AT.	5
4.1 Komendy konfiguracyjne	5
4.1.1 Komenda AT+key	5
4.1.2 Komenda AT+M	5
4.1.3 Komenda AT+InS	6
4.1.4 Komenda AT+InL	6
4.1.5 Komenda AT+InZb	6
4.1.6 Komenda AT+InLam	7
4.1.7 Komenda AT+Out	7
4.1.8 Komenda AT+OutL	8
4.1.9 Komenda AT+OutLID	8
4.1.10 Komenda AT+OutID.....	9
4.1.11 Komenda AT+OutRID	9
4.1.12 Komenda AT+T1	10
4.1.13 Komenda AT+T2	10
4.1.14 Komenda AT+Ret	11
4.1.15 Komenda AT+StepIn	11
4.1.16 Komenda AT+StepMax	12
4.1.17 Komenda AT+OutStep1	12
4.1.18 Komenda AT+OutStep2.....	12
4.1.19 Komenda AT+OutStep3.....	13
4.1.20 Komenda AT+OutStep4.....	13
4.1.21 Komenda AT+OutStep5.....	13
4.1.22 Komenda AT+Init	14
4.1.23 Komenda AT+Time	14
4.1.24 Komenda AT+Save	15
4.1.25 Komenda AT+ActivIn.....	15
4.1.26 Komenda AT+Search.....	15
4.1.27 Komenda AT+Find	16
4.2 Komendy statusowe.....	16

4.2.1 Komenda I	17
4.2.2 Komenda O	17
4.2.3 Komenda R.....	17
4.2.4 Komenda L	18
4.2.5 Komenda AT+Time	18
4.2.6 Komenda T	18
4.2.7 Komenda PING	19
4.3 Komendy sterujące	19
4.3.1 Komenda SetLamp	19
4.3.2 Komenda SetOut	19
4.3.3 Komenda AT+SetTime	20
4.3.4 Komenda AT+SetRol	20
4.3.5 Komenda AT+SMT (SetMaskTime)	20
4.3.6 Komenda AT+RST	21
4.3.7 Komenda PING	22
5 Kontrola, walidacja rozkazów- ERROR	23
6 Wsparcie techniczne.....	24

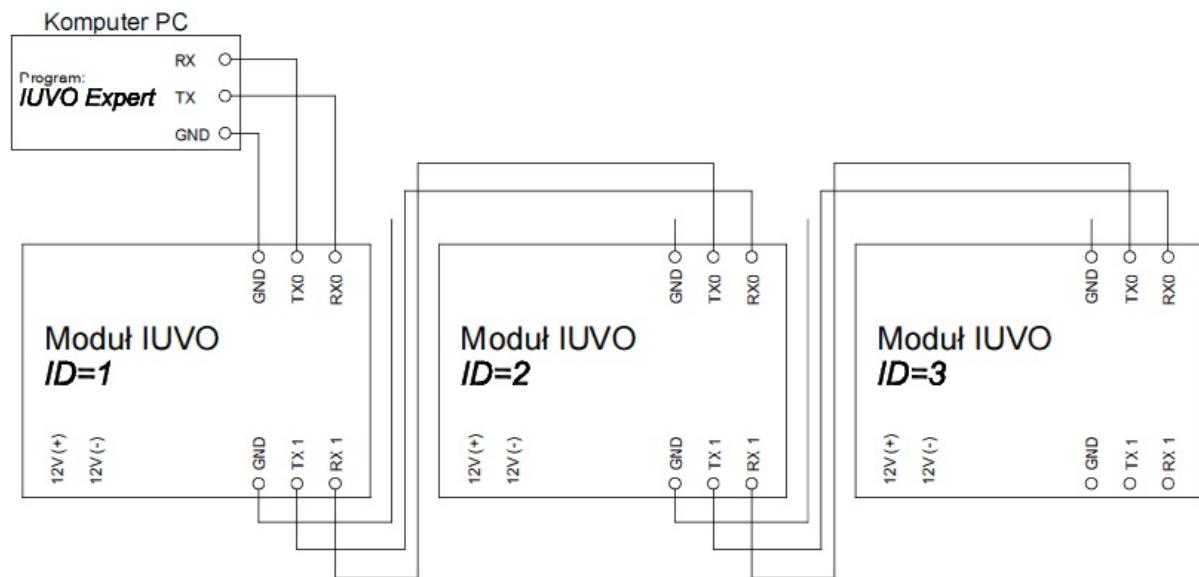


Wersja: 2.23
04.2018

1 Wstęp

Poniższy dokument opisuje protokół komunikacji modułu z modułem, modułem z komputerem klasy PC lub urządzeniem posiadającym port RS232.

Każdy moduł wyposażony jest w dwa złącza komunikacyjne, złącza komunikacyjne pracują w standardzie elektrycznym RS232 (+/-12V). Sposób podłączenia sterowników IUVO w sieć oraz do komputera jest pokazany na rysunku 1.



Rys. 1. Sposób połączenia modułów IUVO w sieć.

2 Ogólny opis protokołu komunikacji

Konfiguracja modułu odbywa się za pomocą wysyłania ciągów znaków- komend AT. Każda z komend zakończona jest znakiem ENTER [CR=13(0x0d)]. Urządzenie wysyłające komendy np komputer klasy PC. Komunikacja z modułem odbywa się w sposób asynchroniczny. Od strony elektrycznej są wykorzystane trzy sygnały RX, TX, GND.

Parametry transmisji:

Prędkość transmisji (Baud Rate): **115200**

Ilość bitów danych (Data bits): **8**

Parzystość (Parity): **brak**

Bity stopu (Stop bits): **1**

Kontrola przepływu : **brak**

4 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, stan lampek itd.)
- **Komendy sterujące**, są to komendy używane do sterowania modułami (zapalenie, zgaszenie, mrugnięcie lampek itd.). Dzięki nim można integrować system IUVO z innymi systemami.

4.1 Komendy konfiguracyjne

4.1.1 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
```

4.1.2 Komenda AT+M

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

opis:

Komenda ustawia tryb pracy funkcji Fx.

ID=1..255 ID konfigurowanego modułu

x=1..16 numer funkcji 1..16

y=1..8- numer trybu 1..7

przykład:

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

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

4.1.3 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
```

4.1.4 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 naciśnięcie 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
```

4.1.5 Komenda AT+InZb

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

opis:

Komenda ustawia czy funkcja Fx, będzie aktywowana przez zwarcie wejścia lub przez rozwarcie wejścia.

ID=1..255 ID konfigurowanego modułu

x=1..16 oznacza numer funkcji

y1=0..1,2 (0: brak akcji,1: aktywacja „zwarcie”, 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
```

4.1.6 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
```

4.1.7 Komenda AT+Out

Moduł IUVO Controller 0806

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

opis:

Komenda ustawia, które wyjścia będą aktywowane przez funkcję Fx

ID=1..255 ID konfigurowanego modułu

x=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 Roller Shutter 0804

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

opis:

Komenda ustawia, które rolety będą aktywowane przez funkcję Fx
ID=1..255 ID konfigurowanego modułu
x=1..16 oznacza numer funkcji
y1=0..1 (Aktywne, Nieaktywne) Roleta 1
y2=0..1 (Aktywne, Nieaktywne) Roleta 2
y3=0..1 (Aktywne, Nieaktywne) Roleta 3
y4=0..1 (Aktywne, Nieaktywne) 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
```

4.1.8 Komenda AT+OutL

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

opis:

Komenda ustawia, które lampki będą zapalane 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 lampkę 4; 5,6,7,8 nie zmienia stanu
```

4.1.9 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/
mrugnięte 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: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=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.
```

4.1.10 Komenda AT+OutID

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

opis:

Komenda (MASKA)ustala jakie wyjścia, zostaną włączone/ wyłączone/ przełączone 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.
```

4.1.11 Komenda AT+OutRID

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

opis:

Komenda (MASKA)ustala jakie rolety, zostaną zamknięte/ otwarte/ zatrzymane/ „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: Otworz, 3: Stop, 4: STEP (Roleta1)

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

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

r4=0..4 / 0:brak reakcji; 1:Zamknij, 2: Otworz, 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 Otwieranie rolety 2 i 3, Zamykanie rolety 4, trybu STEP rolety 1 w module o ID 20.

4.1.12 Komenda AT+T1

Moduł IUVO Controller

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 Roller Shutter 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.

4.1.13 Komenda AT+T2

Moduł IUVO Controller

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 Roller Shutter 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
```

4.1.14 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
```

4.1.15 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
```

4.1.16 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.
```

4.1.17 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
```

4.1.18 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, włącza
wyjścia 2,3; stan pozostałych wyjść zostanie niezmienny
```

4.1.19 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, włącza
wyjścia 2,3,5,6; stan pozostałych wyjść zostanie niezmienny
```

4.1.20 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, włącza
wyjścia 2,3,5,6; stan pozostałych wyjść zostanie niezmienny
```

4.1.21 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
```

4.1.22 Komenda AT+Init

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

opis:

Komenda zachowanie modułu po podaniu napięcia.
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 IUVO Controller
//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 IUVO Controller
//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
```

4.1.23 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.

4.1.24 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.
```

4.1.25 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.

4.1.26 Komenda AT+Search

AT+Search=licznik, ID_modulu<CR>

opis:

Komenda jest wysyłana do modułu w celu nadania adresu ID modułowi.

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 odpowiedź: **AT+Find=5,0605,3<CR>**.

4.1.27 Komenda AT+Find

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

opis:

Odpowiedz na komendę AT+Search,
ID_modulu=1..32 / informuje do którego modułu jest wysłane zapytanie
MAC=0000...FFFF/ numer seryjny modułu
Typ=1.9/

- 1: IUVO Controller0806**
- 2: IUVO Controller0806RTC**
- 3: IUVO RollerShutter**
- 4: IUVO Controller0806T**

:

```
AT+Find= 3,00F1,1<CR>
//Odpowiedz modułu, (typ: IUVO Controller0806) na komendę
AT+Search. ID modułu 3/MAC=00FA
```

Uwagi:

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

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

4.2 Komendy statusowe

Komendy statusowe zwracają aktualny stan modułu. Stan wejść, wyjść oraz lampek. Komendy są wysyłane cyklicznie z częstotliwością 5 sekund, według kolejki.

Przykład:

Jeżeli w sieci znajdują się 3 moduły to informacje statusowe przychodzą w następującej kolejności i sekwencji:

```
Status modułu o ID=1
przerwa 5 sekund
Status Modułu ID=2
przerwa 5 sekund
Status Modułu ID=3
przerwa 5 sekund
Status Modułu o ID=1
przerwa 5 sekund
Status Modułu o ID=2
:
```

Jeżeli Wejście, Wyjście, Lampka zmieni swój stan informacja statusowa zostanie asynchronicznie wygenerowana pod wpływem zdarzenia.

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

0: Wejście nie aktywne

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 (aktywne)

0: Wyjście wyłączone (nieaktywne)

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 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:poniedzialek, 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 modułem

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
```

4.3 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)

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 -> 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
```

4.3.3 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

4.3.4 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
```

4.3.5 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.

4.3.6 Komenda AT+RST

Jest to komenda do restartu modułu oraz informująca o restarcie modułu.

Restartowanie modułu (wysłanie komendy do modułu)

AT+RST=ID_modulu<CR>

opis:

Wysłanie komendy AT+RST=ID_Modułu, spowoduje restart modułu o

ID=ID_Modułu

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_Modułu.

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
```

4.3.7 Komenda PING

Jest to komenda sprawdzenia komunikacji z modułem. Po wysłaniu komendy PING, do modułu, jeżeli komunikacja jest poprawna moduł powinien wysłać komendę PONG.

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

PING=ID_modułu<CR>

opis:

Wysłanie komendy PING=ID_Modułu, spowoduje odesłanie komendy PONG=ID_Modułu

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 PONG=5
```

5 Kontrola, walidacja rozkazów- ERROR

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

ERROR=ID_modulu,ERROR_Ramki,ERROR_Nr<CR>

opis:

ID_modulu- 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

Typ rozkazu w którym wystąpił błąd.	Dokładny opis błędu
ERROR	ERROR=id,0,1<CR> Błędna długość ramki ERROR ERROR=id,0,2<CR> ID nie jest liczbą ERROR=id,0,3<CR> Nieprawidłowy adres ID, spoza zakresu ERROR=id,0,4<CR> parametr error spoza zakresu ERROR=id,0,5<CR> parametr error nie jest liczbą
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> Nieprawidłowy adres ID, spoza zakresu ERROR=id,1,4<CR> parametr out(1..6) spoza zakresu ERROR=id,1,5<CR> parametr out(1..6) nie jest liczbą
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> Nieprawidłowy adres ID, spoza zakresu ERROR=id,2,4<CR> parametr rol(1..4) spoza zakresu ERROR=id,2,5<CR> parametr rol(1..4) nie jest liczbą
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> Nieprawidłowy adres ID, spoza zakresu

	ERROR=id, 3, 4<CR> parametr Lamp(1..8) spoza zakresu ERROR=id, 3, 5<CR> parametr Lamp(1..8) nie jest liczbą
O=	ERROR=id, 4, 1<CR> Błędna długość ramki O= ERROR=id, 4, 2<CR> ID nie jest liczbą ERROR=id, 4, 3<CR> Nieprawidłowy adres ID, spoza zakresu ERROR=id, 4, 4<CR> parametr O(1..8) spoza zakresu ERROR=id, 4, 5<CR> parametr O(1..8) nie jest liczbą
I=	ERROR=id, 5, 1<CR> Błędna długość ramki I= ERROR=id, 5, 2<CR> ID nie jest liczbą ERROR=id, 5, 3<CR> Nieprawidłowy adres ID, spoza zakresu ERROR=id, 5, 4<CR> parametr In(1..8) spoza zakresu ERROR=id, 5, 5<CR> parametr In(1..8) nie jest liczbą
L=	ERROR=id, 6, 1<CR> Błędna długość ramki L= ERROR=id, 6, 2<CR> ID nie jest liczbą ERROR=id, 6, 3<CR> Nieprawidłowy adres ID, spoza zakresu ERROR=id, 6, 4<CR> parametr L(1..8) spoza zakresu ERROR=id, 6, 5<CR> parametr L(1..8) nie jest liczbą
PS=	ERROR=id, 7, 1<CR> Błędna długość ramki PS= ERROR=id, 7, 2<CR> ID nie jest liczbą ERROR=id, 7, 3<CR> Nieprawidłowy adres ID, spoza zakresu ERROR=id, 7, 4<CR> parametr PS(1..8) spoza zakresu ERROR=id, 7, 5<CR> parametr PS(1..8) nie jest liczbą
PL=	ERROR=id, 8, 1<CR> Błędna długość ramki PL= ERROR=id, 8, 2<CR> ID nie jest liczbą ERROR=id, 8, 3<CR> Nieprawidłowy adres ID, spoza zakresu ERROR=id, 8, 4<CR> parametr PL(1..8) spoza zakresu ERROR=id, 8, 5<CR> parametr PL(1..8) nie jest liczbą

6 Wsparcie techniczne

Dokumentację oraz przykłady zastosowania modułów IUVO można znaleźć na stronie:
www.iuvo.it.

W razie dodatkowych pytań, należy skontaktować się telefonicznie lub mailowo:
 e-mail: zasada@iuvo.it
 tel. kom.: +48 510 30 99 70