

Protokół Komunikacyjny TCP/IP+UDP eHouse LAN, WiFi, PRO



Aktualizacja 2016.10.24. Najnowsza wersja do pobrania znajduje się na stronie
<http://www.isys.pl/download/ehouse-lan-protocol-pl.pdf>

Otwarty protokół komunikacji z systemem automatyki budynku eHouse LAN (Ethernet), WiFi, PRO (Ethernet lub WiFi)

System eHouse LAN, WiFi, Pro umożliwia bezpośrednią komunikację ze sterownikiem z zewnętrznego sprzętu i oprogramowania zgodnie z poniższymi protokołami:

- TCP/IP – wysyłanie komend sterujących z autoryzacją lub bez (klient->sterownik eHouse)
- UDP – odbieranie statusów protokołem bezpołączeniowym (Sterownik eHouse->Panele sterujące)

Protokół nadawania komend:

Zastosowano protokół TCP/IP wymagający podłączenia się klienta (Aplikacji PC, Smartfon, Tablet lub inny sprzęt komputerowy) do Serwera (Sterownik eHouse LAN, WiFi, PRO).

Kolejne kroki komunikacji Klient → Serwer:

- 1) Klient nawiązuje połączenie TCP/IP (berkeley sockets) z serwerem na domyślnym porcie 9876 (który może być indywidualnie zmieniony dla całej instalacji eHouse – wszystkich sterowników w systemie – nie zalecane)
- 2) Serwer wysyła 6 bajtów unikalnej odpowiedzi ze aktualnym znacznikiem czasu do autoryzacji ze sterownikiem
- 3) Klient wysyła odpowiedź w następującej ramce:
 - powtórzone 6 bajtów otrzymanych z serwera (index 0..5)
 - 6 bajtów odpowiedzi zależnych od trybu autoryzacji (*): (index 6..11)
 - znak 0x0d (13) (index 12)
 - rozmiar komend - wielokrotność liczby 10 (max 16 zdarzeń) (index 13)
- 4) Klient oczekuje na odpowiedź z serwera:
 - odpowiedź '+' - komenda/zdarzenie zostało przyjęte do kolejki
 - odpowiedź '-' - komenda/zdarzenie nie zostało dodane do kolejki – brak miejsca (**)
 - brak odpowiedzi błąd sieci (**)
- 5) Klient zamyka połączenie wysyłając kod 0x00 do serwera
- 6) Klient rozłącza się z serwera (disconnect)

(**) - należy rozłączyć się i powtórzyć wszystkie kroki po czasie 1-2s.

Metody autoryzacji do sterownika (*):

1. Challenge-Response - dynamiczny kod (zapytanie / odpowiedź) - ze zmiennym kodem
2. XOR Password – dynamicznie szyfrowane hasło (ze zmiennym kodem)
3. Plain Password – proste hasło
4. No Authorisation – brak autoryzacji

Aktywacja konkretnej metody weryfikacji musi być ustawiona w konfiguracji każdego sterownika. Dopuszczalny jest wybrany typ autoryzacji oraz autoryzacje bezpieczniejsze. Np. Wybranie metody autoryzacji Challenge-Response blokuje wszystkie pozostałe metody, Wybranie autoryzacji XOR Password umożliwia stosowanie: XOR Password oraz Challenge-Response.

Challenge-Response: dynamicznie zmieniany kod (unikalny dla każdej instalacji)

- Klient odbiera odpowiedź z serwera po podłączeniu klienta (6B)
- Klienta wysyła powtórzoną odpowiedź z serwera (6B) oraz zaszyfrowaną odpowiedź (6B) w następującej formie (odpowiedź Servera **XOR** Unikalny kod instalacji eHouse obliczony z niepowtarzalnych danych). Zarówno kod serwera jak i odpowiedź jest unikalny i zawiera znacznik czasu i komendy z dużym opóźnieniem są ignorowane . Ten tryb jest dostępny tylko w oryginalnych aplikacjach eHouse: **eHouse.exe (RS-485)**, **eHouseWiFi.exe (LAN, WiFi)** oraz **eHousePRO**. W przypadku dużych instalacji i umów licencyjnych możliwe jest wygenerowanie indywidualnego algorytmu uwzględniającego Kod Licencyjny (**Vendor Code**) unikalny i przyznawany dla każdego Partnera. Pełny algorytm autoryzacji nie jest udostępniany, aby nie zmniejszać bezpieczeństwa systemu eHouse i umożliwiać włamania hakerom.

XOR Password: dynamicznie szyfrowane hasło (ze zmiennym kodem):

- Klient odbiera odpowiedź z serwera po podłączeniu klienta (6B)
- Klienta wysyła powtórzoną odpowiedź z serwera (6B) oraz zaszyfrowaną odpowiedź (6B) w następującej formie (Odpowiedź Serwera **XOR** Hasło dla danej instalacji). Zarówno kod serwera jak i odpowiedź jest unikalny i zawiera znacznik czasu i komendy z dużym opóźnieniem są ignorowane . Hasło nie jest przesyłane bezpośrednio przez łącza komunikacyjne (LAN, WiFi, Internet). Ten tryb umożliwia tworzenie własnego oprogramowania i bezpieczną autoryzację do sterownika w aplikacjach eHouse: eHouse Java, Android oraz indywidualnych aplikacjach klienta z dostępem przez internet włącznie. Ten algorytm autoryzacji jest udostępniany klientom oraz kody źródłowe w wielu językach programowania znajdują się na stronie <http://www.isys.pl/download/>.

Uproszczony przykład algorytmu:

```

unsigned char ServerChallenge[6];           //Server response after connection of client
unsigned char Password[6];                 //password for verification 6 bytes
unsigned char ClientResponse[170];         //response to server from client
unsigned char CommandToSend[10];           //command/event to send to server
int socket;                                //bsd socket
unsigned char buff[20];                    //receive of client buffer
recv(socket,ServerChallenge,6,0);
memcpy(ClientResponse,ServerChallenge,6); //copy ServerChallenge to client response
for (char i=0;i<6;i++)                      //hashing client response with password
{
    ClientResponse[6+i]=ServerChallenge[i]^Password[i];
}
ClientResponse[12]=13; //static field
ClientResponse[13]=10; //size of one command (10B)
memcpy((unsigned char *)&ClientResponse[14],CommandToSend,10); //copy command code
send(socket,ClientResponse,14+ClientResponse[13],0); //send to BSD socket
buff[0]=0;
if (recv(socket,buff,1,0)<0) //error receiving response
{
    close(socket);
    return -1;
}
else
    if (buff[0]=='+')
        { //event received by server confirmation OK
        }
    else //no confirmation of event/command
        { //should be retried
        }
buff[0]=0; //Send Close command and disconnect
send(socket,buff,1,0);
flush(socket);
Delay100ms;
close(socket);

```

Plain Password / Proste hasło:

- Klient odbiera Odpowiedź z serwera po podłączeniu klienta (6B)
- Klienta wysyła powtórzoną Odpowiedź z serwera (6B) oraz hasło Hasło dla danej instalacji (6B). Zarówno kod serwera jak i odpowiedź są unikalne jednak hasło jest przesyłane bezpośrednio przez łącza komunikacyjne (LAN, WiFi, Internet). Sprawdzany jest znacznik czasu, aby wyeliminować pakiety powtórzone i retransmisje oraz proste próby sabotażu. Ten tryb umożliwia tworzenie własnego oprogramowania i prostą autoryzację do sterownika w indywidualnych aplikacjach klienta oraz XOR-Password w aplikacjach eHouse: eHouse Java, Android. Należy unikać go przy stosowaniu na łączach publicznych (Internet). Ten algorytm autoryzacji jest udostępniany klientom oraz kody źródłowe w wielu językach programowania znajdują się na stronie <http://www.isys.pl/download/>.

Uproszczony przykład algorytmu:

```

unsigned char ServerChallenge[6];           //Server response after connection of client
unsigned char Password[6];                 //password for verification 6 bytes
unsigned char ClientResponse[170];        //response to server from client
unsigned char CommandToSend[10];          //command/event to send to server
int socket;                                //bsd socket
unsigned char buff[20];                    //receive of client buffer
recv(socket,ServerChallenge,6,0);          //receive Challenge from server
memcpy(ClientResponse,ServerChallenge,6); //copy ServerChallenge to client response
for (char i=0;i<6;i++)                     //adding plain password
{
    ClientResponse[6+i]=Password[i];       //plain password without encryption
}
ClientResponse[12]=13;                     //static field
ClientResponse[13]=10;                     //size of one command (10B;
memcpy((unsigned char *)&ClientResponse[14],CommandToSend,10); //copy command code
send(socket,ClientResponse,14+ClientResponse[13],0); //send to BSD socket
buff[0]=0;
if (recv(socket,buff,1,0)<0) //error receiving response
{
    close(socket);
    return -1;
}
else
    if (buff[0]=='+')
        { //event received by server confirmation OK
        }
    else //no confirmation of event/command
        { //should be retried
        }
buff[0]=0; //Send Close command and disconnect
send(socket,buff,1,0);
flush(socket);
Delay100ms;
close(socket);

```

No Authorization / Brak autoryzacji do sterownika:

- 1) Klient odbiera Odpowiedź z serwera po podłączeniu klienta (6B)
- 2) Klienta wysyła powtórzoną Odpowiedź z serwera (6B) oraz dowolne (6B), aby zachować długość ramki komunikacyjnej. Dane przechodzące przez łącza komunikacyjne (LAN, WiFi, Internet) nie są sprawdzane przez serwer. Ten tryb umożliwia testowanie własnego oprogramowania w początkowej fazie bez logowania się do sterownika. Zezwala także na prostą autoryzację (PLAIN-Password) do sterownika w indywidualnych aplikacjach klienta oraz XOR-Password w aplikacjach eHouse: eHouse Java, Android. Nie należy go stosować w instalacjach produkcyjnych szczególnie z dostępem przez Internet. Jest on udostępniany klientom wraz kody źródłowe w wielu językach programowania znajdują się na stronie

<http://www.isys.pl/download/>.

Uproszczony przykład algorytmu:

```

unsigned char ServerChallenge[6];           //Server response after connection of client
unsigned char Password[6];                 //password for verification 6 bytes
unsigned char ClientResponse[170];         //response to server from client
unsigned char CommandToSend[10];          //command/event to send to server
int socket;                                //bsd socket
unsigned char buff[20];                    //receive of client buffer
recv(socket,ServerChallenge,6,0);          //receive Challenge from server
//memcpy(ClientResponse,ServerChallenge,6); //copy ServerChallenge to client response
//for (char i=0;i<6;i++)                    //adding plain password
//    {
//        ClientResponse[6+i]=Password[i]; //plain password without encryption
//    }
//ClientResponse[12]=13; //static field*/
for (char i=0;i<13;i++) {ClientResponse[6+i]=0;} //anything here to keep proper frame length
ClientResponse[13]=10; //size of one command (10B;
memcpy((unsigned char *)&ClientResponse[14],CommandToSend,10); //copy command code
send(socket,ClientResponse,14+ClientResponse[13],0); //send to BSD socket
buff[0]=0;
if (recv(socket,buff,1,0)<0) //error receiving response
    {
        close(socket);
        return -1;
    }
else
    if (buff[0]=='+')
        { //event received by server confirmation OK
        }
    else //no confirmation of event/command
        { //should be retried
        }
    buff[0]=0; //Send Close command and disconnect
    send(socket,buff,1,0);
    flush(socket);
    Delay100ms;
    close(socket);
    
```

Protokół wysyłania zdarzeń/komend dla systemu eHouse

Standardowe zdarzenia/komendy systemu eHouse (DirectEvent) posiadają długość 10B (bajtów)

Znaczenie kolejnych bajtów zdarzenia:

- 1) ADRH - Address H – wysoka składowa adresu sterownika (MSB)
- 2) ADRL - Address L – niska składowa adresu sterownika (LSB)
- 3) komenda/kod zdarzenia
- 4) argument 1 dla komendy
- 5) argument 2 dla komendy
- 6) argument 3 dla komendy
- 7) argument 4 dla komendy
- 8) argument 5 dla komendy
- 9) argument 6 dla komendy
- 10) argument 7 dla komendy

Ze względu na częstą potrzebę integracji różnych wariantów systemu **eHouse** zastosowano standaryzację i uproszczenia, które dla własnej wygody powinno się wykorzystywać. Dotyczy to szczególnie adresów przyznawanych sterownikom oraz standardowych portów eHouse. Ze względu na długotrwały rozwój systemu od roku 2000 i potrzebę łączenia różnych interfejsów komunikacyjnych konieczne było zastosowanie standaryzacji, aby najnowsze wersje mogły automatycznie obsługiwać standardy zastosowane we wczesnych wersjach systemu umożliwiając ich rozwój i rozbudowę po latach.

Chociaż adresy sterowników można nadawać dowolne składowe ADRH, ADRL z przedziału (1..254) 0x01..0xfe należy unikać zmiany składowej ADRH, gdyż jest ona wykorzystana przez aplikacje systemu eHouse do wykrycia wariantu sterownika (**RS-485, CAN, RF, PRO, LAN, WiFi**).

ADR H (1, 2, 55) – eHouse RS-485

ADR H (0x7f..0x8f) – eHouse CAN

ADR H (0x70..0x7e) – eHouse RF

Pozostałe adresy ADRH - eHouse LAN, WiFi, PRO (ten sam adres wspólny, aby wszystkie sterowniki pracowały w masce **255.255.255.0**. Adres IP jest tworzony w następujący sposób

192.168.ADRH.ADRL

Dla sterowników LAN, WiFi, PRO także składowa ADRL ma znaczenie:

ADR L (100..199) – dla sterowników eHouse WiFi

ADR L (200) – eHouse PRO

ADR L (201..249) – eHouse LAN (EthernetRoomManager'y)

ADR L (250..254) – eHouse LAN (CommManager'y)

Komendy/zdarzenia zależą od typu architektury eHouse (LAN, PRO, CAN/RF, WiFi, RS-485) oraz typu sterownika (CommManager, RoomManager, HeatManager, małe sterowniki)

Komendy zawarte w niniejszym dokumencie dotyczą tylko sterowników eHouse LAN (CommManager, LevelManager, EthernetRoomManager).

Zdarzenia eHouse LAN: – zależy od wariantu sterownika LM, CM, ERM oraz wersji firmwaru

Nazwa	Kod Decymalny	Znaczenie
EXECPROFILE	2	execute output program change
SETLIGHT	4	set dimmer
SETADC	12	set adc levels
SETOUTPUT	33	change output state
IRALIAS	39	send ir command
INCDECPROFILE	96	Increment / decrement Program
ADC_PROGRAM	97	set adc program
SECŪ_PROGRAM	98	run rollers program & zone – CM only
ZONE_CHANGE	99	zone change - CM only
ROLLERSSINGLE	100	single rollers execution – CM only
ROLLERSMULTI	101	rollers program execution – CM only
BATCH_ROM	104	run multiple events from rom bank
MODIFY_LIGHTS	105	Change Dimmers Settings
SETSINGLELIGHT	106	Set single Dimmer Value
SET_TIME	107	Obsolete
SETDMX	108	Set DMX Single Chanel
SETDMXRGB	109	Set DMX 3 channels
SENDDALICMD	242	Set DALI Single Chanel
SETDALI	243	Set DALI Single Chanel
SETDALIRGB	244	Set DALI RGB
MODADC	245	Modify ADC Levels
SETDALISCENE	246	Set DALI Program/scene
SETDMXSCENE	247	Set DMX Program/scene
EVENT_SUBMIT_TCP_STATUS	248	send DATA to log analyzer if connected
EVENT_SEND_TCP	249	send DATA to opened TCP Query connection
EVENT_SEND_UDP	250	broadcast DATA via udp
EVENT_SEND_TO_LOG	251	send DATA to log analyzer if connected
SENDIRCOMMAND	252	send infrared code (RC Control)
SCAN_IR	253	scan IR remote controller code
RESETDEV	254	reset device

Numer bajtu (0..9) oraz znaczenie

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	CMD	Arg 1	Arg 2	Arg 3	Arg 4	Arg 5	Arg 6	Arg 7

Adres Sterownika Komenda Argumenty specyficzne dla danej komendy systemu eHouse

EXECPROFILE	2	execute output program change
-------------	---	-------------------------------

Uruchomienie programu wyjść on/off i wyjść ściemniaczy LED

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	2	Arg 1	0	0	0	0	0	0

CMD - 2

Arg 1 – numer programu/sceny świetlnej 0..23



SETLIGHT	4	Set Dimmer
----------	---	------------

Ustawienie stanu ściemniaczy LED/RGB 3 kanały

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	4	Arg 1	Arg 2	Arg 3	Arg 4	0	0	0

CMD = 4

Arg 1 – nr startowy kanału

Arg 2 – poziom ściemniacza R (0..100)

Arg 3 – poziom ściemniacza G (0..100)

Arg 4 – poziom ściemniacza B (0..100)

SETADC	12	Set ADC levels
--------	----	----------------

Ustawienie progów dolnego i górnego dla wejścia pomiarowego ADC (regulacja)

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	12	Arg 1	Arg 2	Arg 3	Arg 4	Arg 5	0	0

CMD - 12

Arg 1 – numer wejścia pomiarowego ADC 1..16

Arg 2 – wyższa składowa poziomu dolnego MSB

Arg 3 – niższa składowa poziomu dolnego LSB

Arg 4 – wyższa składowa poziomu górnego MSB

Arg 5 - wyższa składowa poziomu górnego LSB

Wartości progów dolnego i górnego oblicza się następująco:

$ADCLow = (Arg2 \ll 8) + Arg3$

$ADCHigh = (Arg4 \ll 8) + Arg5$

Wartość ADCLow, ADCHigh progów dolnych w postaci bezwzględnej wskazań przetwornika ADC (A/C) 10 bitowego $<0..1024$) zależy od zastosowanego czujnika oraz skali odwzorowania.

Dla standardowych czujników (Pomiar napięcia, czujnik temperatury, oświetlenia sugerujemy, aby wykorzystać dane z aplikacji **eHouseWiFi.exe**).

SETOUTPUT	33	execute output change
-----------	----	-----------------------

Ustawienie pojedynczego wyjścia on/off

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	33	Arg 1	Arg 2	Arg 3	Arg 4	0	0	0

CMD – 33 lub 1

Arg 1 – numer wyjścia 1..128

Arg 2 – stan wyjścia

0 – wyłącz

1 – włącz

2 – przełącz (stan przeciwny)

Arg 3 – Timeout LSB – mniej znaczący bajt

Arg 4 – Timeout MSB – bardziej znaczący bajt

Timeout=(Arg4<<8)+Arg3

Timeout – wartość w sekundach po którym wyjście automatycznie się wyłączy

IRALIAS	39	Send IR Command
---------	----	-----------------

Wysłanie makra 4 kodów podczerwieni IR z bazy danych sterownika

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	39	Arg 1	Arg 2	Arg 3	Arg 4	0	0	0

CMD – 39

Arg 1 – index pierwszego kodu podczerwieni z konfiguracji sterownika

Arg 2 – index drugiego kodu podczerwieni z konfiguracji sterownika

Arg 3 – index trzeciego kodu podczerwieni z konfiguracji sterownika

Arg 4 – index czwartego kodu podczerwieni z konfiguracji sterownika



INCDECPROFILE	96	Increment/decrement Program
---------------	----	-----------------------------

Zmniejszenie / zwiększenie numeru programu / sceny świetlnej

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	96	Arg 1	0	0	0	0	0	0

CMD - 96

Arg 1 – Kierunek zmiany programu

- 1 – Zwiększenie o 1 numeru programu
- 2 – Zmniejszenie o 1 numeru programu



ADC_PROGRAM	97	execute ADC/measurement program change
-------------	----	--

Uruchomienie programu wejść pomiarowych / regulacyjnych

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	97	Arg 1	0	0	0	0	0	0

CMD – 97

Arg 1 – numer programu pomiarowo/regulacyjnego 0..11



SECU_PROGRAM	98	execute security/rollers program change
--------------	----	---

Uruchomienie programu napędów rolet wraz ze strefą zabezpieczeń (tylko CM)

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	98	Arg 1	0	0	0	0	0	0

CMD - 98

Arg 1 – numer programu napędów 0..23



ZONE_CHANGE	99	execute security zone change
--------------------	-----------	-------------------------------------

Zmiana strefy/maski zabezpieczeń tylko (CM)

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	99	Arg 1	Arg 2	0	0	0	0	0

CMD - 99

Arg 1 – numer strefy zabezpieczeń 0..23

Arg 2 – opóźnienie 0 - brak, pozostałe ~30s

ROLLERSINGLE	100	execute drive/servo change
--------------	-----	----------------------------

Uruchomienie napędu rolety/serwa (tylko CM)

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	100	Arg 1	Arg 2	0	0	0	0	0

CMD - 100

Arg 1 – numer napędu $n=(0..35)$

Arg 2 – kierunek ruchu:

0 – brak zmiany

1 – aktywacja linii/wyjścia nr $(n*2+1)$ / **Otwórz**

2 – aktywacja linii/wyjścia nr $(n*2+2)$ / **Zamknij**

3 – **STOP**

Somfy - aktywacja obu wyjść $(n*2+1)$ i $(n*2+2)$

DIRECT - dezaktywacja obu wyjść $(n*2+1)$ i $(n*2+2)$

Sterowanie napędu odbywa się 2 liniami (wyjściami) o numerach $(n*2+1)$ oraz $(n*2+2)$. W konfiguracji sterownika CM musi być ustawiony tryb pracy wszystkich wyjść jako napędy (Somfy lub Direct).

Uwaga: niewłaściwe ustawienie trybu pracy napędów Somfy/Direct może spowodować uszkodzenie napędów. W przypadku trybu Somfy występuje sterowanie 2 kierunkami jednocześnie (STOP).

ROLLERSMULTI	101	execute multiple drives change
--------------	-----	--------------------------------

Jednoczesne uruchomienie wielu napędów rolet/serw (tylko CM)

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	101	Arg 1	Arg 2	Arg 3	Arg 4	Arg 5	Arg 6	Arg 7

Arg 1..7 – ustawienia linii napędów – 1 bit dla każdego wyjścia/kierunku) znaczenie bitowe

ArgX (1..7) => n =X-1 (0..6) - bity

Bit [MSb..LSb]	7	6	5	4	3	2	1	0
Nr wyjścia	8n+8	8n+7	8n+6	8n+5	8n+4	8n+3	8n+2	8n+1
Napęd	N/4+4	N/4+4	N/4+3	N/4+3	N/4+2	N/4+2	N/4+1	N/4+1
Kierunek	Close	Open	Close	Open	Close	Open	Close	Open

Sterowanie napędu odbywa się 2 liniami (wyjściami) o numerach (x*2+1) oraz (x*2+2) gdzie x-nr napędu. W konfiguracji sterownika CM musi być ustawiony tryb pracy wszystkich wyjść jako napędy (Somfy lub Direct).

Uwaga: niewłaściwe ustawienie trybu pracy napędów Somfy/Direct może spowodować uszkodzenie napędów. W przypadku trybu Somfy występuje sterowanie 2 kierunkami jednocześnie (STOP).



BATCH_ROM	104	execute batch of commands/events
-----------	-----	----------------------------------

Uruchomienie makra zdarzeń zapisanych w pamięci zdarzeń sterownika

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	104	Arg 1	Arg 2	0	0	0	0	0

CMD - 104

Arg 1 – Numer zdarzenia z pamięci makr (0..max)

Arg 2 – ilość zdarzeń do uruchomienia (1..max)

MODIFY_LIGHS	105	Modify Dimmers Settings
---------------------	------------	--------------------------------

Zmiana ustawień ściemniaczy LED i DMX (tylko ERM)

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	105	Arg 1	Arg 2	Arg 3	0	0	0	0

CMD - 105

Arg 1 – numer ściemniacza 1..3 (wbudowane LED) oraz 4..24 (zewnętrzne DMX)

Arg 2 – operacja

0 – włączenie zmniejszania poziomu jasności (do zatrzymania lub osiągnięcia minimalnej wartości poziomu) z krokiem zmiany w polu **Arg 3**

1 – włączenie zwiększenia poziomu jasności (do zatrzymania lub osiągnięcia maksymalnej wartości poziomu) z krokiem zmiany w polu **Arg 3**

2 – stop (zatrzymanie zmiany)

3 – ustawienie wartości z pola **Arg 3**

4 – przełącz - jeśli ściemniacz w ruchu zatrzymuje się, jeśli zatrzymany uruchamia zmianę w przeciwnym kierunku do ostatniej zmiany

5 – zmniejsza jednorazowo poziom ściemniacza o wartość w polu **Arg 3**

6 – zwiększa jednorazowo poziom ściemniacza o wartość w polu **Arg 3**

Arg 3 – wartość kroku przy zmianie lub poziom ustawienia



SETSINGLELIGHT	106	Set single dimmer light level
-----------------------	------------	--------------------------------------

Ustawienie poziomu ściemniacza wewnętrznego LED lub zewnętrznego DMX

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	106	Arg 1	Arg 2	0	0	0	0	0

CMD - 106

Arg 1 – nr ściemniacza (1..max) 1..3 (wewnętrzne), 4..max zewnętrzne ściemniacze DMX

Arg 2 – poziom ściemniacza (0..100%)

SETDMX	108	Set DMX dimmer settings
--------	-----	-------------------------

Zmiana ustawień ściemniacza DMX

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	108	Arg 1	Arg 2	Arg 3	0	0	0	0

CMD - 108

Arg 1 – numer ściemniacza (3..Max)

Arg 2 – operacja

0 – włączenie zmniejszania poziomu ściemniacza o wartość w polu **Arg 3**

1 – włączenie zwiększania poziomu ściemniacza o wartość w polu **Arg 3**

2 – zatrzymanie zmiany poziomu ściemniacza

3 – ustawienie poziomu ściemniacza, wartość w polu **Arg 3**

Arg 3 – wartość kroku przy zmianie lub poziom ustawienia

SETDMXRGB	109	Set DMX RGB Dimmer
-----------	-----	--------------------

Ustawienie ściemniacza DMX / RGB (tylko ERM)

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	109	Arg 1	Arg 2	Arg 3	Arg 4	0	0	0

CMD - 109

Arg 1 – n numer ściemniacza DMX/RGB * 3 kanały (1..Max)

Arg 2 – poziom ściemniacza R (czerwony)

Arg 3 – poziom ściemniacza G (zielony)

Arg 4 – poziom ściemniacza B (niebieski)

Uwaga: Aby poniższa funkcja działała prawidłowo konieczne jest umieszczenie ściemniaczy DMX RGB na początku szeregu, aby ściemniacz DMX/RGB znajdował się pod adresem

DMX=3*(n-1)+1

SENDDALICMD	242	Send DALI Command
--------------------	------------	--------------------------

Wysłanie komendy DALI na magistralę oświetlenia DALI

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	242	Arg 1	Arg 2	0	0	0	0	0

CMD - 242

Arg 1 – komenda MSB (najstarszy bajt)

Arg 2 – komenda LSB (najmłodszy bajt)

SETDALI	243	Set DALI dimmer
---------	-----	-----------------

Ustawienie ściemniacza DALI

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	243	Arg 1	Arg 2	Arg 3	0	0	0	0

CMD - 243

Arg 1 – numer kanału DALI (1..Max)

Arg 2 – operacja

0 – obniżenie poziomu jasności o wartość w polu **Arg 3**

1 – zwiększenie poziomu jasności o wartość w polu **Arg 3**

3 - ustawienie poziomu jasności na wartość w polu **Arg 3**

Arg 3 – krok zmiany lub poziom ściemniacza



SETDALISCENE	246	Set DALI light scene/program
--------------	-----	------------------------------

Uruchomienie programu /sceny świetlnej oświetlenia DALI

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	246	Arg 1	0	0	0	0	0	0

CMD - 246

Arg 1 – numer programu/sceny świetlnej DALI (1..Max)



SETDMXSCENE	247	Set DMX light scene/program
-------------	-----	-----------------------------

Uruchomienie programu /sceny świetlnej oświetlenia DMX

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	247	Arg 1	0	0	0	0	0	0

CMD - 247

Arg 1 – numer programu/sceny świetlnej DMX (1..Max)



EVENT_SUBMIT_TCP_STATUS	248	Submit data with TCP/IP & UDP Status
-------------------------	-----	--------------------------------------

Dołączenie danych do statusu sterownika

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	248	Arg 1	Arg 2	Arg 3	Arg 4	Arg 5	Arg 6	Arg 7

CMD – 248

Arg 1..7 – Dane binarne do wysłania razem ze statusem sterownika (w polu STATUS_MORE)



EVENT_SEND_TCP	249	Send Data as Log to TCP/IP Logger
----------------	-----	-----------------------------------

Wysyła 7 bajtów danych do podłączonej aplikacji Logów TCP

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	249	Arg 1	Arg 2	Arg 3	Arg 4	Arg 5	Arg 6	Arg 7

CMD - 249

Arg 1 ..7 – 7 bajtów danych do wysłania do podłączonej aplikacji **TCPllogger**



EVENT_SEND_UDP	250	Send/broadcast Data as log via UDP protocol
-----------------------	------------	--

Wysyła 7 bajtów danych po protokole UDP

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	250	Arg 1	Arg 2	Arg 3	Arg 4	Arg 5	Arg 6	Arg 7

CMD - 250

Arg 1 ..7 – 7 bajtów danych do wysłania po protokole UDP



EVENT_SEND_TO_LOG	251	Send Data to Log
--------------------------	------------	-------------------------

Wysyła 7 bajtów danych do logu

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	251	Arg 1	Arg 2	Arg 3	Arg 4	Arg 5	Arg 6	Arg 7

CMD - 251

Arg 1 ..7 – 7 bajtów danych do wysłania do logu



SENDIRCOMMAND	252	Send IR Command
----------------------	------------	------------------------

Wysłanie kodu podczerwieni

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	252	Arg 1	Arg 2	Arg 3	Arg 4	Arg 5	Arg 6	Arg 7

CMD - 252

Arg 1 – Typ standardu podczerwieni (przechwycony w aplikacji eHouseWiFi)

Arg 2..7 – Kod pilota podczerwieni (przechwycony w aplikacji eHouseWiFi) MSB..LSB

SCAN_IR	253	SCAN IR Command
---------	-----	-----------------

Przechwycenie i nauka kodu podczerwieni

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	253	0	0	0	0	0	0	0

CMD - 253

RESETDEV	254	Reset Device
-----------------	------------	---------------------

Restart sterownika

0	1	2	3	4	5	6	7	8	9
ADRH	ADRL	254	0	0	0	0	0	0	0

CMD - 254

Dekodowanie statusu sterowników eHouse LAN

Statusy sterowników eHouse LAN są wysyłane po protokole UDP automatycznie na porcie 6789 bezpośrednio do wszystkich urządzeń dostępnych w sieci LAN. Port ten może zostać zmieniony inny jednak nie zalecamy zmiany tego portu o ile nie jest to konieczne. Status sterowników może zostać być wysyłany do podłączonego panela TCP/IP (klienta).

Ramka Statusu / Frame

Size [1B]	AddrH [1B]	AddrL [1B]	Type [1B]	Date [Size B]	Checksum [2B]
			s	Status	
			l	Log Data	

Indeksy bajtów statusu:

Pole	[Index]	{rozmiar}
STATUS_SIZE	[0]	{1B}
STATUS_ADDRH	[1]	{1B}
STATUS_ADDRL	[2]	{1B}
STATUS_CODE	[3]	// 's' {1B}
STATUS_DIMMERS	[4]	{20B}
STATUS_DMX_DIMMERS	[STATUS_DIMMERS+3]	{17B}
STATUS_ADC_LEVELS	[24]	{32B}
STATUS_MORE	[56]	//Additional DATA {8B}
STATUS_ADC2_LEVELS	[64]	//8*2*MSb, 2 Most Significant Bits {8B}
STATUS_ADC_ETH	[72]	//ADC Measurements results {16 * 2B}
STATUS_ADC_ETH_END	[STATUS_ADC_ETH+32] [104]	
STATUS_OUT_I2C	[STATUS_ADC_ETH_END]	//CM only {20B}
STATUS_OUT	[STATUS_ADC_ETH_END] [104]	//ERM only {5B}
STATUS_DMX_DIMMERS2	[STATUS_OUT_I2C+5] [109]	//ERM only {15B}
STATUS_DALI2	[STATUS_OUT_I2C+5]	//ERM only {15B}
STATUS_INPUTS_I2C	[STATUS_OUT_I2C+20] [124]	//CM only {12B}
STATUS_INPUTS	[STATUS_INPUTS_I2C] [124]	//ERM only {2B}
STATUS_DIMMERS3	[STATUS_INPUTS_I2C+2] [126]	//ERM only {22B}
STATUS_DALI	[STATUS_INPUTS_I2C+2] [126]	//ERM only {22B}
STATUS_ALARM_I2C	[STATUS_INPUTS_I2C+12] [136]	//CM only {12B}
STATUS_WARNING_I2C	[STATUS_ALARM_I2C+12] [148]	//CM only {12B}
STATUS_MONITORING_I2C	[STATUS_WARNING_I2C+12] [160]	//CM only {12B}
STATUS_PROGRAM_NR	[STATUS_MONITORING_I2C+12] [172]	{1B}
STATUS_ZONE_NR	[STATUS_PROGRAM_NR+1] [173]	//CM only {1B}
STATUS_ADC_PROGRAM	[STATUS_ZONE_NR+1] [174]	{1B}

eHouse LAN ANY Controllers / Wszystkie sterowniki

eHouse LAN CommManager (CM) ONLY

eHouse LAN EthernetRoomManager (ERM) ONLY

Bits Location in STATUS_OUT, STATUS_INPUTS fields Pola bitowe ERM

Index / Bit Locations	7	6	5	4	3	2	1	0
STATUS_OUT	O8	O7	O6	O5	O4	O3	O2	O1
STATUS_OUT+1	O16	O15	O14	O13	O12	O11	O10	O9
STATUS_OUT+2	O24	O23	O22	O21	O20	O19	O18	O17
STATUS_OUT+3	O32	O31	O30	O29	O28	O27	O26	O25
STATUS_OUT+4						O35	O34	O33
STATUS_INPUTS	IN8	IN7	IN6	IN5	IN4	IN3	IN2	IN1
STATUS_INPUTS+1					IN12	IN11	IN10	IN9

Analogiczne pola bitowe dla CM

STATUS_OUT_I2C + (0..20) - status wyjść cyfrowych on/off

STATUS_INPUTS_I2C + (0..12) - status wejść cyfrowych on/off

STATUS_ALARM_I2C + (0..12) - status wejść cyfrowych w stanie alarmu (syrena)

STATUS_WARNING_I2C + (0..12) - status wejść w stanie ostrzeżenia (światło ostrzegawcze)

STATUS_MONITORING_I2C + (0..12) - status wejść w stanie aktywnego monitoringu