Grenton

Instrukcja użytkownika

Grenton 2

Wersja dokumentu: 1.0.10

Data: 01.12.2022

Spis treści

Ważne informacje

I. Struktura systemu

II. Fundamenty - interfejs logiczny GRENTON

1. Wprowadzenie
2. Cechy
    - 2.1. Cechy wbudowane
    - 2.2. Cechy użytkownika
3. Metody
4. Zdarzenia
5. Adresy cech i metod

III. Przygotowanie projektu

1. Przygotowanie instalacji elektrycznej
2. Wybór architektury systemu
3. Zasilanie modułów

IV. Instalacja komponentów

1. Montaż modułów na rozdzielni
2. Montaż modułów podtynkowych kablowych
3. Montaż modułów podtynkowych Z-Wave

V. Object Manager

1. Instalacja OM
    - A. Windows
    - B. macOS
    - C. Linux
2. Struktura OM
    - 2.1. Filtrowanie obiektów
    - 2.2. Zmiana nazwy obiektu
3. Pliki projektów
    - 3.1. Katalog zapisanych projektów
    - 3.2. Kopia zapasowa projektu (backup)
4. Podstawowe elementy
    - 4.1. Konfigurator obiektów
    - 4.2. Script builder

4.3. Diagram połączeń
4.4. Visual Builder
4.5. myGrenton
4.6. Kosz

VI. Podstawowa konfiguracja systemu

1. Połączenie OM z CLU
2. Adresy IP
3. Utworzenie nowego projektu
4. Funkcja CLU Discovery
    4.1. Dodanie modułów do projektu
    4.2. Podmiana / przypisywanie modułów podczas Discovery
5. Statusy CLU
    5.1. Diody modułu
    5.2. Ikona modułu CLU w OM
6. Obsługa modułów Z-Wave
    6.1. Dodawanie modułów Z-Wave
    6.2. Usuwanie modułów Z-Wave
    6.3. Brak komunikacji z modułem Z-Wave - mechanizm zliczania niepowodzeń komunikacji i blokowania komunikacji urządzenia w sieci Z-Wave
    6.4. Wskazówki konfiguracyjne sieci Z-Wave
    6.5. Czyszczenie informacji o węzłach
7. Wysyłanie konfiguracji do CLU
8. Wartości początkowe cech
9. Tworzenie podstawowych powiązań
10. Przeprowadzanie aktualizacji
    10.1. Proces aktualizacji bazy interfejsów
    10.2. Proces aktualizacji firmware'u CLU
    10.3. Proces aktualizacji firmware'u modułów z serii 2.0
    10.4. Status CLU / modułów w oknie aktualizacji firmware'u
    10.5 Procedura wymuszania aktualizacji modułu
11. Widok diagnostyczny
    11.1 Konfiguracja widoku diagnostycznego
12. Inne operacje na systemie

VII. Zaawansowane funkcje konfiguracyjne

1. Kontenery
2. Skrypty
    2.1 Tworzenie skryptów w trybie graficznym
    2.2. Tworzenie skryptów w trybie tekstowym
    2.3. Parametry skryptu
    2.4. Wywoływanie skryptów
    2.4. Funkcja Znajdź / Zastąp
    2.5. Kopiowanie skryptów
3. Data i czas

VIII. Visual Builder

1. Sterowanie systemem z poziomu smartfonów
2. Struktura interfejsu
3. Aplikacja na smartfony – GRENTON HOME MANAGER
4. Tworzenie nowego interfejsu
    4.1. Wybór skórki graficznej
    4.2. Tworzenie stron interfejsu
    4.3. Komponenty
    4.4. Panele
    4.5. Kontenery
    4.6. Wstawianie komponentów i tworzenie powiązań z obiektami systemu
    4.7. Wysyłanie interfejsu do urządzenia mobilnego
5. Automatyczne tworzenie interfejsu - generator GUI

5.1. Tworzenie interfejsu z dostępną rozdzielczością
5.2. Tworzenie interfejsu z własną rozdzielczością
5.3. Zmiana orientacji interfejsu z własną rozdzielczością
6. Konfiguracja wideodomofonu
6.1. Podłączenie i konfiguracja wideodomofonu
6.2. Stworzenie i konfiguracja interfejsu aplikacji
6.3. Wykonanie połączenia z domofonu
7. Obsługa obrazu z kamer
8. Zdalny dostęp aplikacji mobilnej do systemu
8.1. Konfiguracja systemu
8.2. Ustawienie przekierowania portów w routerze sieci lokalnej
8.3. Konfiguracja aplikacji mobilnej Home Manager
8.4. Uruchomienie zdalnego dostępu

IX. Obiekty CLU

1. Timer
2. Kalendarz
3. Harmonogram
4. Regulator PID
5. Termostat
6. Push
7. Czujnik obecności
8. Kalendarz wschodów i zachodów słońca
9. Harmonogram zdarzeń

X. Pomiar mediów

1. Wirtualny pomiar mediów
1.1. Uruchomienie wirtualnego pomiaru mediów po stronie Object Managera
1.2 Używanie pomiaru mediów po stronie aplikacji Home Manager
2. Rzeczywisty pomiar mediów
2.1. Ustawienia rzeczywistego pomiaru mediów po stronie Object Managera

XI. Funkcje serwisowe CLU

1. Przywracanie ustawień fabrycznych CLU - *Hard Reset*
2. Diagnostyka systemu - *Zapisz paczkę diagnostyczną*

XII. SMART PANEL

1. Wyposażenie Smart Panelu
2. Podłączenie Smart Panelu do CLU
3. Informacje pomocne przy tworzeniu konfiguracji
4. Konfiguracja modułu Smart Panel w wersji v3
4.1. Parametry konfiguracyjne
4.2. Tworzenie konfiguracji przycisków i wyświetlacza
4.3. Tworzenie konfiguracji sensora gestów
4.4. Konfiguracja czujnika zbliżenia
4.5. Tworzenie konfiguracji wielu stron panelu dotykowego
5. Konfiguracja modułu Smart Panel w wersji v4
5.1. Parametry konfiguracyjne
5.2. Tworzenie konfiguracji sensora gestów
5.3. Konfiguracja czujnika zbliżenia
5.4. Obiekt Panel - nowa funkcjonalność
5.5. Obiekt Panel - mechanizm zarządzania stronami
5.6. Kompatybilność wstecz
5.7. Tworzenie konfiguracji z wykorzystaniem obiektu strony Buttons
5.8. Tworzenie konfiguracji z wykorzystaniem obiektu strony FreeDraw
5.9. Tworzenie konfiguracji z wykorzystaniem obiektu strony Thermostats
5.10. Łączenie obiektów w większe przyciski
6. Konfiguracja modułu Smart Panel w wersji v6
6.1. Parametry konfiguracyjne
6.2. Nowa funkcjonalność

6.3. Zmiana UI oraz mechanizmu działania stron typu Thermostats

XIII. Moduł GATE ALARM

1. Informacje ogólne
2. Konfiguracja modułu
3. Integracja z centralą Satel
    3.1. Informacje ogólne
    3.2. Konfiguracja dla systemu Satel
    3.3. Obiekty wirtualne
4. Integracja z centralą Jablotron
    4.1. Informacje ogólne
    4.2. Konfiguracja dla systemu Jablotron
    4.3. Obiekty wirtualne
5. Obiekt wirtualny - Timer
6. Przywracanie ustawień fabrycznych - *Hard Reset*
7. Parametry konfiguracyjne

XIV. Moduł GATE MODBUS

1. Informacje ogólne
2. Konfiguracja modułu
    2.1. Ustawianie czasu za pomocą serwera NTP
    2.2. Konfiguracja obiektu wirtualnego Modbus
    2.3. Konfiguracja obiektu wirtualnego ModbusValue
3. Parametry rejestrów
    3.1. Obiekt wirtualny Modbus
    3.2. Obiekt wirtualny ModbusValue
4. Przywracanie ustawień fabrycznych - *Hard Reset*
5. Parametry konfiguracyjne

XV. Moduł GATE HTTP

1. Informacje ogólne
2. Konfiguracja modułu
2.1. Obiekty wirtualne
    2.1.1. HTTPRequest
    2.1.2. Pobieranie określonych wartości z otrzymanej odpowiedzi (XML,JSON)
    2.2.1. HttpListener
    2.2.2. Przygotowanie odpowiedzi wysyłanej do serwera
    2.2.3. Odczyt wartości kluczy z parametru querystringparams
    2.3.1. Timer
3. Możliwość połączenia z Gate za pomocą TELNET
4. Kompleksowa integracja z systemami zewnętrznymi przy użyciu urządzenia GATE HTTP
    4.1. System
    4.2. Sterowanie wyjściem
    4.3. Pobieranie stanu
    4.4. Kolejność zdarzeń
    4.5. Synchronizacja zdarzeń
    4.6. Potwierdzenie zwrotne
    4.7. Timeout
    4.8. Wiele obiektów
    4.9. Stan dla złożonego systemu
    4.10. Push Notyfikacje
5. Przywracanie ustawień fabrycznych - *Hard Reset*
6. Parametry konfiguracyjne

XVI. Moduł DALI Controller

1. Informacje ogólne
2. Konfiguracja modułu
3. Obiekty

XVII. Moduły Z-Wave

1. Fibaro UBS
    1.1. Informacje ogólne
    1.2. Obiekty
2. NEO Coolcam Motion Sensor (PIR)
    2.1. Informacje ogólne
    2.2. Obiekty
3. NEO Coolcam Door / Window Sensor
    3.1. Informacje ogólne
    3.2. Obiekty
4. INFIBITY Motion Sensor (PIR) [NEO Coolcam]
    4.1. Informacje ogólne
    4.2. Obiekty
5. INFIBITY Door/Window Sensor [NEO Coolcam]
    5.1. Informacje ogólne
    5.2. Obiekty
6. INFIBITY Water Sensor [NEO Coolcam]
    6.1. Informacje ogólne
    6.2. Obiekty
7. Heiman Smart Smoke Sensor
    7.1. Informacje ogólne
    7.2. Obiekty
8. INFIBITY Siren Alarm [NEO Coolcam]
    8.1. Informacje ogólne
    8.2. Obiekty
9. Danfoss Living Connect
    9.1. Informacje ogólne
    9.2. Obiekty
10. POPP Z-Weather
    10.1. Informacje ogólne
    10.2. Obiekty
11. FAKRO AMZ Solar
    11.1. Informacje ogólne
    11.2. Obiekty
12. FAKRO ARF
    12.1. Informacje ogólne
    12.2. Obiekty
13. FAKRO FTP_V
    13.1. Informacje ogólne
    13.2. Obiekty
14. Fibaro RGBW
    14.1. Informacje ogólne
    14.2. Obiekty
15. Remotec ZXT-120
    15.1. Informacje ogólne
    15.2. Opis konfiguracji urządzenia
    15.3. Obiekty
16. Remotec ZXT-310
    16.1. Informacje ogólne
    16.2. Opis konfiguracji urządzenia
    16.3. Obiekty
17. Aeotec Nano Switch
    17.1. Informacje ogólne
    17.2. Obiekty
18. Aeotec Dual Nano Switch
    18.1. Informacje ogólne
    18.2. Obiekty
19. Aeotec Nano Dimmer
    19.1. Informacje ogólne

19.2. Obiekty
20. Aeotec Nano Shutter
20.1. Informacje ogólne
20.2. Obiekty
21. Aeotec Nano Shutter (V2)
21.1. Informacje ogólne
21.2. Obiekty

XVIII. Aplikacja mobilna myGrenton

1. Instalacja oraz pierwsze uruchomienie aplikacji myGrenton
1.1. Instalacja
1.2. Pierwsze uruchomienie, interfejs demonstracyjny
2. Tworzenie interfejsu
2.1. Dodawanie strony do interfejsu
2.2. Usuwanie strony z interfejsu
2.3. Kopiowanie interfejsu
3. Widgety
3.1. Nagłówek (HEADER)
3.2. Wartość (VALUE)
3.3. Wartość v2 (VALUE_V2)
3.4. Wartość Podwójny (VALUE_DOUBLE)
3.5. Włącz/Wyłącz (ON_OFF)
3.6. Włącz/Wyłącz Podwójny (ON_OFF_DOUBLE)
3.7. Scena (SCENE)
3.8. Scena Podwójny (SCENE_DOUBLE)
3.9. Ściemniacz (DIMMER)
3.10. Ściemniacz v2 (DIMMER_V2)
3.11. Oświetlenie LED (LED)
3.12. Termostat (THERMOSTAT)
3.13. Rolety (ROLLER_SHUTTER)
3.14. Rolety v2 (ROLLER_SHUTTER_V2)
3.15. Rolety v3 (ROLLER_SHUTTER_V3)
3.16. Kamera (CAMERA)
3.17. Tekst (TEXT)
3.18. Harmonogram (SCHEDULER)
3.19. Harmonogram zdarzeń (EVENT_SCHEDULER)
3.20. Multisensor (MULTISENSOR)
3.21. Pilot TV (TV_REMOTE_CONTROL)
3.22. Pilot Audio (AUDIO_REMOTE_CONTROL)
3.22. Kontaktron (CONTACT_SENSOR)
3.23. Kontaktron Podwójny (CONTACT_SENSOR_DOUBLE)
3.24. Suwak (SLIDER)
3.25. Personalizacja widgetu
3.26. Usuwanie widgetu
3.27. Kopiowanie widgetu
4. Personalizacja interfejsu
4.1. Zmiana nazwy interfejsu
4.2. Zmiana ikony interfejsu
4.3. Zmiana koloru interfejsu
4.4. Blokowanie dostępu przez chmurę
5. Wysyłanie interfejsu na urządzenie
5.1. Wysłanie interfejsu myGrenton poprzez kod QR lub manualnie
5.2. Udostępnienie interfejsu myGrenton poprzez chmurę
6. Ustawienia aplikacji oraz interfejsu
6.1. Ustawienia aplikacji
6.2. Ustawienia interfejsu

XIX. Rozproszenie Logiki Grenton 2.0

1. Konfiguracja trybu Rozproszenia Logiki

1.1. Działanie Logiki Rozproszonej pomiędzy obiektami DIN oraz wyjściowymi
1.2. Działanie Logiki Rozproszonej pomiędzy obiektami BUTTON oraz wyjściowymi
1.3. Działanie Logiki Rozproszonej pomiędzy obiektami PANEL_PAGE z przypisanymi PANEL_BUTTON oraz wyjściowymi
2. Default Mode
2.1. Default Mode dla modułów wejściowych oraz modułów wyjściowych
2.2. Default Mode dla modułów posiadających własne wejścia / wyjścia
3. Przywrócenie komunikacji pomiędzy CLU a modułem

Ważne informacje

Uwaga! Niniejsza dokumentacja obejmuje funkcjonalności oraz zasady działania dla modułów Grenton z serii 2.0. Zachowany zostaje opis funkcjonalny dla Object Managera a także Home Managera. Aplikacja myGrenton jest całkowicie kompatybilna z systemami Grenton 2.0 - w przypadku systemów 1.0 dostęp do poszczególnych funkcji może być ograniczony lub całkowicie niedostępny.

I. Struktura systemu

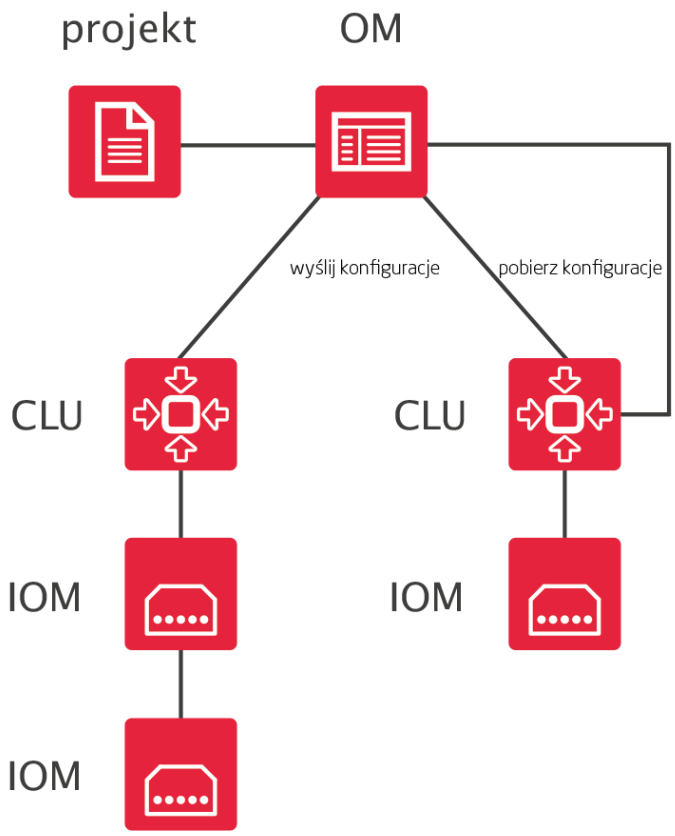
System Inteligentnych budynków GRENTON został zaprojektowany z myślą o obsłudze małych, średnich i dużych obiektów. Instalacje stworzone na bazie systemu można w łatwy sposób modyfikować, rozbudowywać oraz integrować z innymi systemami.

W skład systemu wchodzą: moduły CLU, moduły IOM, Object Manager, sensory oraz aplikacje na smartfony:

- Moduły CLU (*Common Logic Unit*). Realizują funkcję przetwarzania logiki oraz przechowywania konfiguracji. CLU stanowią podstawę każdego systemu. CLU komunikują się ze sobą za pomocą magistrali systemowej, która działa w oparciu o standard Ethernet 100 Mbps. Moduły CLU zapewniają również komunikację z modułami IOM przy pomocy magistrali lokalnej.
- Moduły IOM są modułami realizującymi funkcje wejścia/wyjścia, które podłączane są do CLU poprzez magistralę lokalną TF-Bus lub bezprzewodowo z wykorzystaniem standardu Z-Wave. Moduły IOM mogą zawierać różnego typu wejścia/wyjścia (np. przekaźniki, włączniki, sensory światła, temperatury) oraz ich kombinacje.
- Object Manager – jest oprogramowaniem umożliwiającym konfigurację systemu, funkcji logicznych itd.
- Aplikacje sterujące – pozwalają na uruchamianie zaprojektowanych w OM graficznych interfejsów użytkownika, które dają możliwość sterowania funkcjami systemu przy użyciu smartfonów, tabletów, komputerów PC, TV itp.

Konfiguracja systemu przechowywana jest w postaci pliku projektu i ustawiana przy pomocy programu Object Manager (OM). Utworzona konfiguracja jest następnie przesyłana do modułów CLU, które przechowują ją w swojej pamięci. Moduły IOM nie przechowują konfiguracji i są sterowane bezpośrednio z CLU, do którego zostały podłączone.

W razie utraty pliku projektu OM posiada funkcję ściągnięcia danych z CLU. Pobranie danych z CLU wiąże się jednak z utratą: widoku graficznego utworzonych skryptów, kontenerów, interfejsów mobilnych oraz typów obiektów (źródło/odbiornik).


projekt
OM
wyślij konfiguracje
pobierz konfiguracje
CLU
CLU
IOM
IOM
IOM

II. Fundamenty - interfejs logiczny GRENTON

1. Wprowadzenie

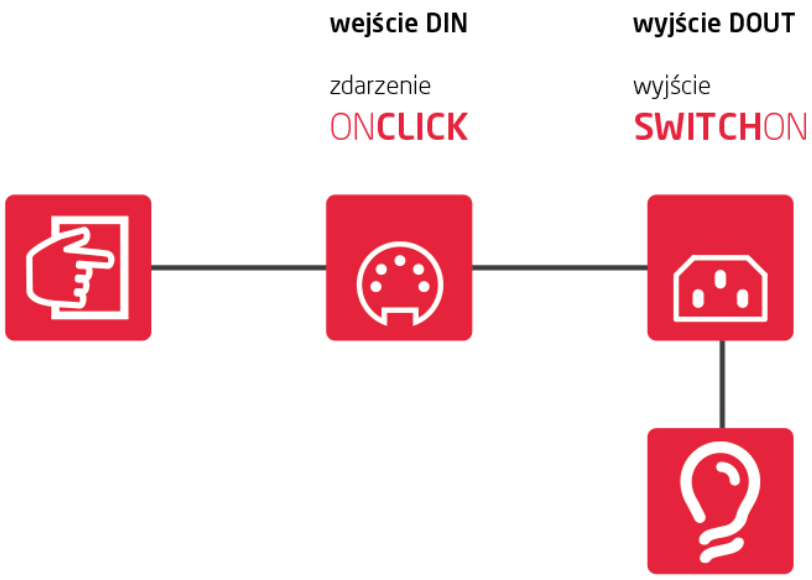
System GRENTON działa w oparciu o tzw. model wywoływany zdarzeniami (event driven model). Domownicy i otoczenie powodują generowanie w systemie zdarzeń, z którymi powiązane są reakcje systemu (np. w odpowiedzi na naciśnięcie wyłącznika system włącza lampę).

Podstawą interfejsu logicznego są obiekty. W systemie GRENTON, każdy obiekt jest traktowany oraz zachowuje się w taki sposób jak każdy fizyczny przedmiot, np. piłka. Każdy obiekt posiada swoje cechy, możemy wykonać na nim pewne czynności, może również wywoływać jakieś zdarzenia. Odnosząc to do naszej piłki: możemy nią kopnąć, czyli sterować, tocząca się piłka może przewrócić stojącą butelkę, a więc wywoła zdarzenie i na koniec możemy powiedzieć, że jest czerwona, czyli posiada pewne swoje cechy.

W systemie każde wejście lub wyjście posiada własny zbiór cech, metod i zdarzeń, które nazywane są jego interfejsem logicznym.

Unikalnym rozwiązaniem systemu GRENTON jest to, iż każda cecha lub metoda są dostępne w dowolnym miejscu systemu, na każdym CLU, niezależnie od tego, na którym CLU, wejściu lub wyjściu fizycznie się one znajdują. Możliwe jest więc wywołanie metody z wyjścia podłączonego do CLU A w następstwie zdarzenia, które zaszło w obrębie CLU B.

Również wyjścia posiadają specyficzne dla siebie zdarzenia, co umożliwia np. zaświecenie jednej lampy wskutek włączenia innej. Pełną listę zdarzeń metod i cech każdego wejścia/wyjścia można znaleźć w karcie katalogowej modułu.



2. Cechy

2.1. Cechy wbudowane

Cechy wbudowane to zbiór parametrów/informacji opisujących dany obiekt (wejście, wyjście itd.). Niektóre z cech mogą być ustawiane w trakcie działania systemu i służą do określenia sposobu działania obiektu (tryb pracy przycisku). Inne natomiast mogą być jedynie odczytywane, gdyż np. pokazują fizyczne wartości (np. cecha `Value` dla czujnika temperatury).

2.2. Cechy użytkownika

W CLU można zdefiniować cechy, które mogą być następnie użyte jako zmienne do przechowywania parametrów podczas działania systemu (np. liczniki, znaczniki). Cechy użytkownika mogą być używane w identyczny sposób jak cechy wbudowane, z tą różnicą, że wszystkie cechy użytkownika mogą być zarówno zapisywane do pamięci ulotnej, jak i z niej odczytywane.

3. Metody

Metody są rozkazami, które mogą zostać wydane danemu obiektowi. Każdy obiekt posiada charakterystyczne dla siebie metody. Dla wyjścia przekaźnikowego mogą być to metody `SwitchOn` (załącz) lub `SwitchOff` (wyłącz). Dodatkowo metody mogą zawierać parametry wymagane lub opcjonalne, które precyzują sposób ich wywołania (np. czas włączenia).

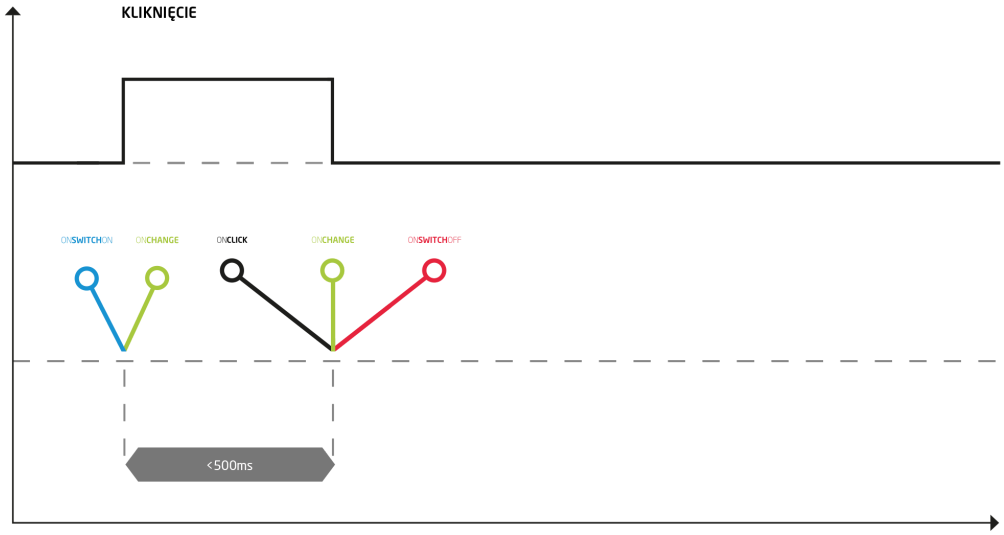
4. Zdarzenia

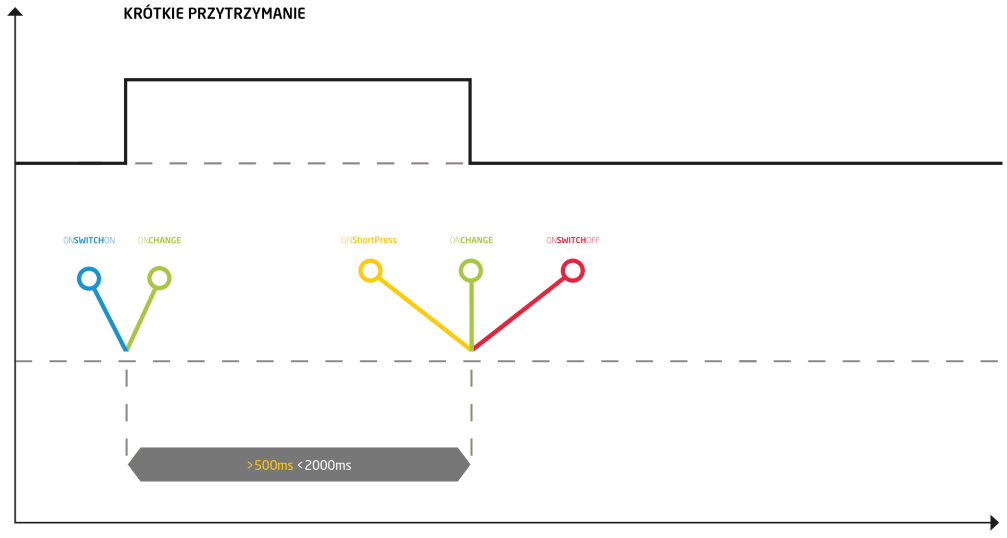
Zdarzenia są elementami interfejsu logicznego wywoływanymi w reakcji na zmiany zachodzące w związku z obiektem (np. włączenie przycisku, zmiana temperatury itp.). Z każdym zdarzeniem możemy powiązać nawet kilka metod, które zostaną wykonane w momencie zaistnienia zdarzenia, np. w momencie naciśnięcia przycisku zostaną włączone lampy. Wiążąc zdarzenia jednych obiektów (głównie z wejść, ale czasem również wyjść) z metodami innych obiektów, tworzymy konfigurację logiczną systemu.

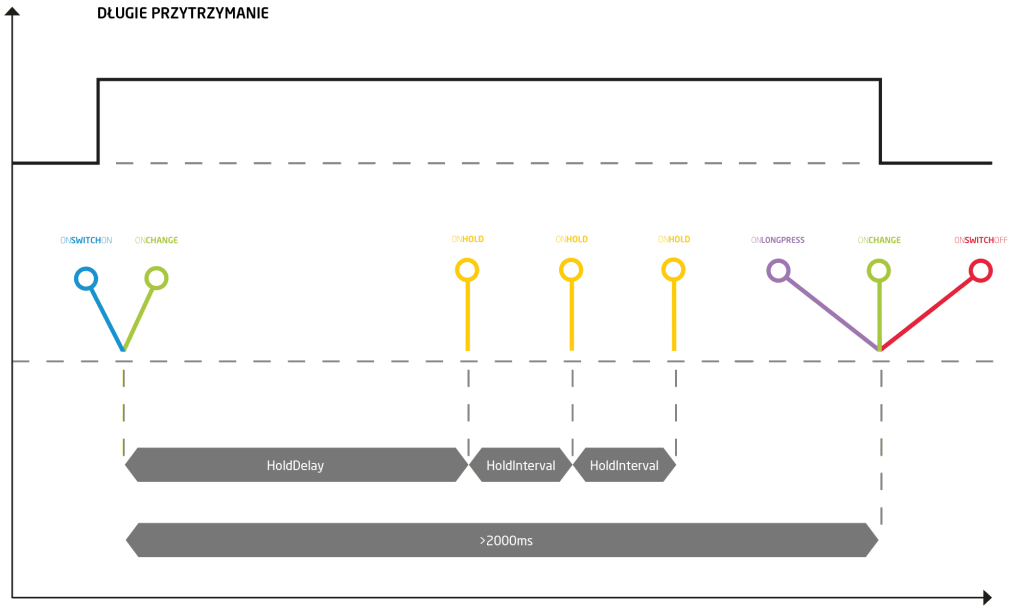
Każdy typ obiektu (rodzaj wejścia/wyjścia) posiada własna listę zdarzeń, które są wywoływane w ściśle określony sposób, w zależności od akcji podejmowanych przez użytkownika. Przykładowo wejście binarne posiada następująca listę zdarzeń:

- `OnChange`
- `OnSwitchOn`
- `OnSwitchOff`
- `OnShortPress`
- `OnLongPress`
- `OnClick`
- `OnHold`

które wywoływane są zgodnie z poniższymi schematami:


KLIKNIĘCIE
ONSWITCHON
ONCHANGE
ONCLICK
ONCHANGE
ONSWITCHOFF
<500ms


KRÓTKIE PRZYTRZYMANIE
ONSWITCHON
ONCHANGE
ONShortPress
ONCHANGE
ONSWITCHOFF
>500ms <2000ms



5. Adresy cech i metod

Każda cecha i metoda posiada w systemie adres, dzięki któremu można odwoływać się do niej w skryptach oraz podczas powiązywania ze zdarzeniami. Adres składa się z 3 części połączonych znakami `->`:

- identyfikator CLU lub kontenera;
- nazwa obiektu (wejścia, wyjścia, CLU);
- nazwa cechy/metody wraz z parametrami (o ile występują).

Przykładowo: `CLU1->Lampa1->SwitchOn()` – metoda powodująca włączenie wyjścia `Lampa1`

`Oświetlenie->Lampa1->value()` – cecha pokazująca, czy lampa jest włączona czy wyłączona, dla lampy znajdującej się w kontenerze *Oświetlenie*.

III. Przygotowanie projektu

1. Przygotowanie instalacji elektrycznej

> **Uwaga!** Instalacje elektryczne w domach mieszkalnych oraz budynkach użyteczności publicznej mogą być wykonywane tylko zgodnie z obowiązującymi przepisami i normami elektrycznymi oraz tylko przez wykwalifikowanych specjalistów posiadających odpowiednią wiedzę oraz wymagane uprawnienia wynikające z przepisów prawa obowiązującego w danym kraju.

A. Topologia instalacji elektrycznej

System GRENTON umożliwia tworzenie instalacji zarówno scentralizowanych jak i rozproszonych. Dla nowo projektowanych budynków sugerujemy sprowadzenie wszystkich obwodów do jednej rozdzielni elektrycznej, co pozwoli na bardziej elastyczne podejście w projektowaniu instalacji i bardziej ekonomiczne zarządzanie zasobami.

Do każdego urządzenia, które ma zostać połączone z systemem, powinien zostać wydzielony osobny obwód elektryczny zakończony w rozdzielni elektrycznej. Dobór przekroju przewodów powinien zostać przeprowadzony w oparciu o obowiązujące normy. Jeśli nie ma możliwości ułożenia przewodu bezpośrednio od rozdzielni do sterowanego urządzenia, możliwe są trzy ścieżki:

1. Wyniesienie modułu CLU wraz z modułami IOM. W takim przypadku moduły CLU należy połączyć w rozdzielni z modułem wyniesionym za pomocą magistrali systemowej – takie rozwiązanie ma sens, gdy w ramach jednego systemu integrujemy co najmniej dwa budynki.
2. Wyniesienie jednego lub więcej modułów IOM. Łączenie modułów odbywa się przy pomocy magistrali lokalnej – rozwiązanie polecane, gdy wynoszona jest niewielka liczba modułów.
3. Wykorzystanie radiowych modułów IOM działających w oparciu o Z-Wave. Rozwiązanie sugerowane w sytuacji, gdy nie ma możliwości wykonania instalacji kablowej (istniejące budynki itp.).

B. Magistrala

W systemie znajdują się 2 magistrale:

1. **Systemowa**, służąca do połączenia pomiędzy modułami CLU-CLU oraz CLU-SMARTFONY itp.

Magistrala systemowa – Ethernet. Moduły mogą być ze sobą łączone szeregowo.

Maksymalna długość kabla pomiędzy dwoma modułami CLU wynosi 90 m. Zalecany kabel

UTP (minimum kat. 5e). Długość magistrali systemowej może zostać zwiększona przez wykorzystanie urządzeń sieciowych - np. switch, router, itp.

2. **Lokalna**, służąca do połączenia pomiędzy modułami CLU-IOM.

   Magistrala lokalna – moduły IOM mogą być łączone ze sobą za pomocą tasiemek, a także dołączane do magistrali przy pomocy modułu BUS MODULE. Magistrala musi być ułożona szeregowo. Maksymalna długość magistrali pomiędzy dwoma jej końcami wynosi 300 m.

> **Uwaga!** Może być konieczne dodatkowe zasilanie magistrali.

Zalecany kabel o stałej impedancji falowej i przekroju minimalnym 0,5 mm², np. kabel UTP (opcjonalnie ekranowane kable: FTP lub E-BUS). Przy większej ilości modułów lub bardziej rozbudowanej magistrali należy uwzględnić spadki napięć przy doborze przekroju kabla magistralnego.

C. Przydatne wskazówki

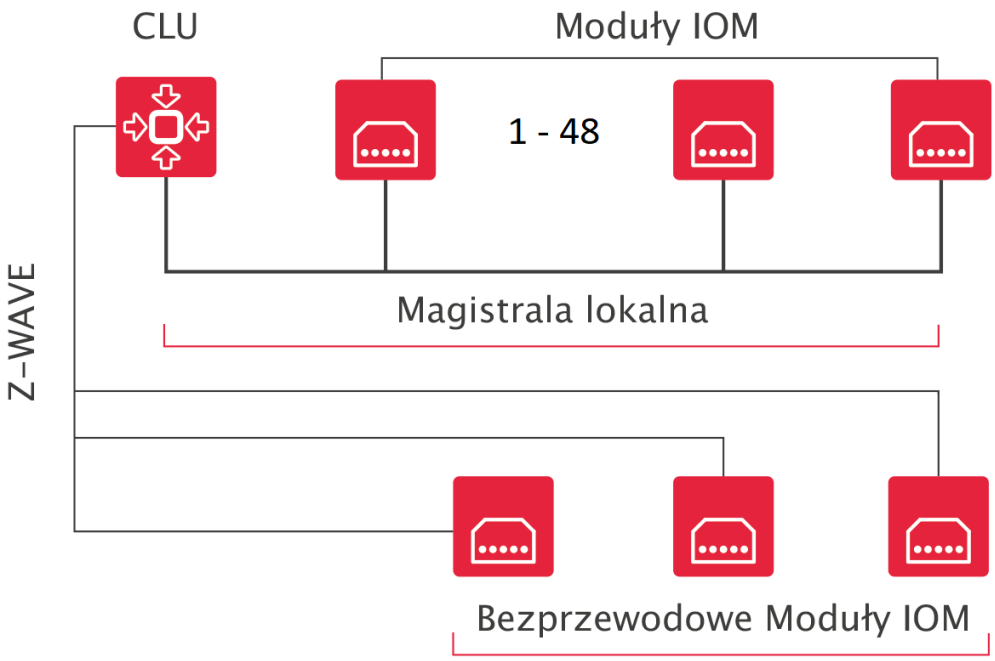
- Przed rozpoczęciem wykonania instalacji elektrycznej należy przygotować projekt instalacji systemu inteligentnego.
- Jeśli jeszcze nie wiadomo, którymi urządzeniami będzie odbywało się sterowanie, zaleca się dociągnięcie okablowania do wszystkich możliwych miejsc.
- Do włączników oświetlenia można stosować dowolny cienki przewód, np. YTDY – pozwoli to zmniejszyć koszty instalacji.
- Należy pamiętać o przygotowaniu instalacji pod czujniki temperatury i stację pogodową.
- Umieszczenie gniazda zasilania na tarasie i poprowadzenie do niego osobnego zasilania da możliwość sterowania zasilaniem w tym gnieździe z systemu.

2. Wybór architektury systemu

W zależności od rodzaju obiektów, wymagań oraz wielkości można stosować różne konfiguracje – system jest w pełni skalowalny. W zależności od skali i potrzeb możliwych jest kilka konfiguracji:

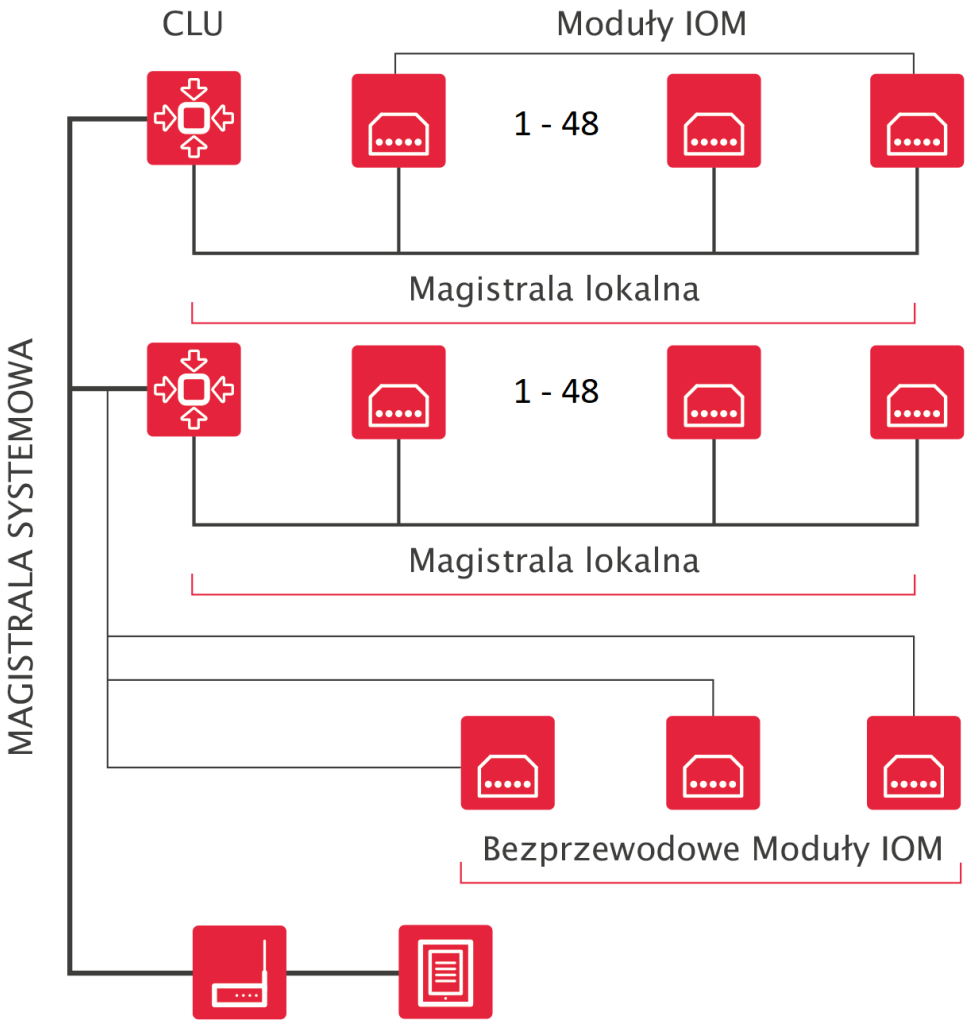
A. Konfiguracja podstawowa – scentralizowany System z jednym CLU

Na schemacie został przedstawiony system zbudowany w oparciu o jedno CLU. W systemie skonfigurowanym w taki sposób maksymalnie może znajdować się do 48 modułów IOM bez względu na ich rodzaj (lub do 400 obiektów). Należy pamiętać o zapewnieniu magistrali zasilania odpowiedniego do obciążenia.



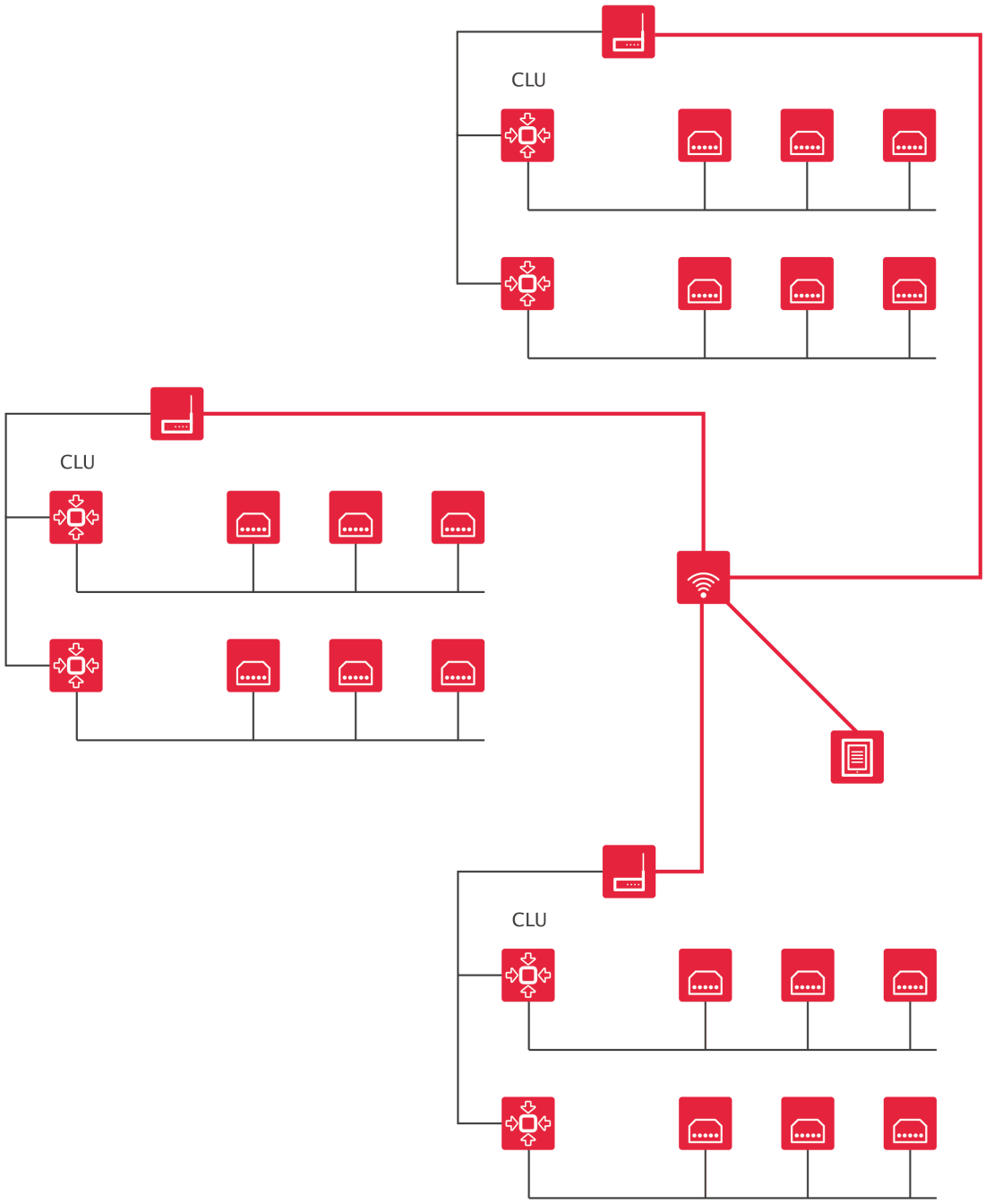
B. Konfiguracja rozbudowana - system rozproszony z wieloma CLU i sterowaniem z tabletu

Pojemność systemu można zwiększyć poprzez dołączanie kolejnych modułów CLU. Jednostki CLU łączymy pomiędzy sobą za pomocą magistrali systemowej. Instalacja dodatkowo może być rozszerzona o smartfony, tablety itp.



C. Integracja wielu budynków w jeden system

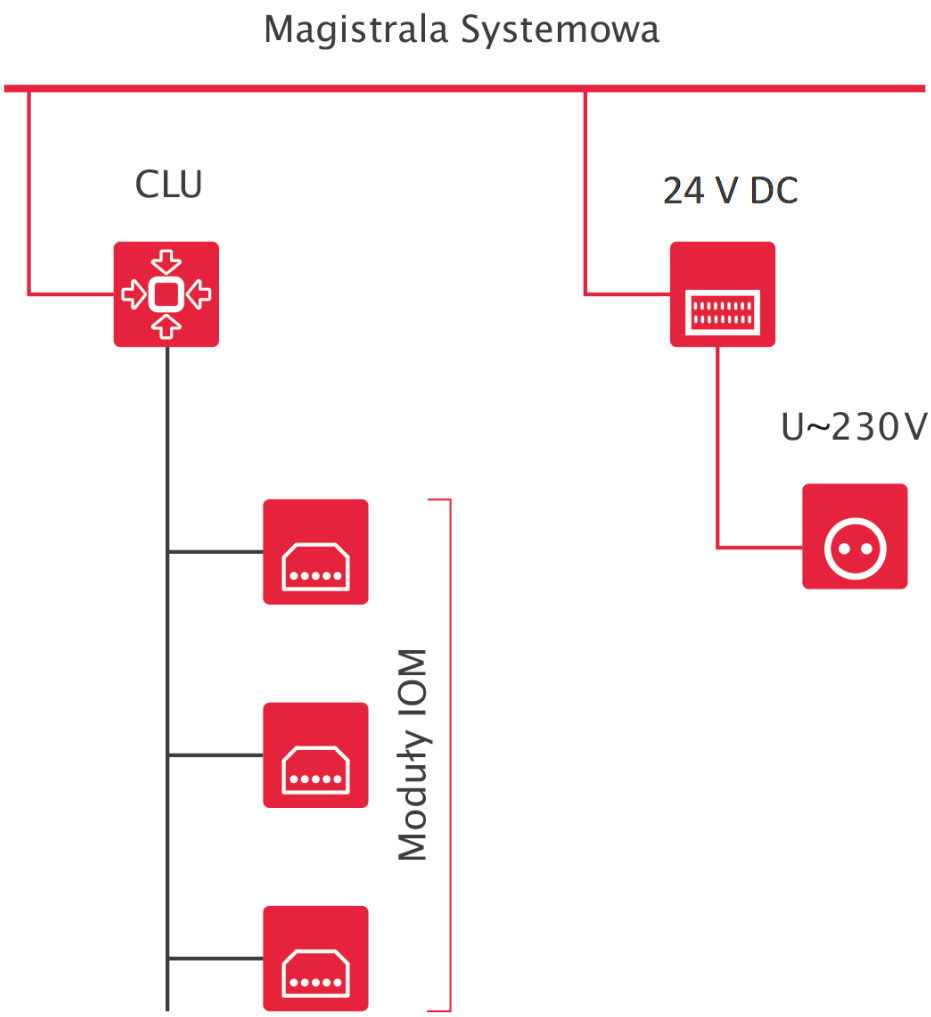
System może być rozbudowywany praktycznie w nieograniczony sposób. Do jednego systemu może zostać wpięte kilka obiektów. Dzięki czemu będziemy mieć możliwość centralnego sterowania za pomocą jednego systemu.



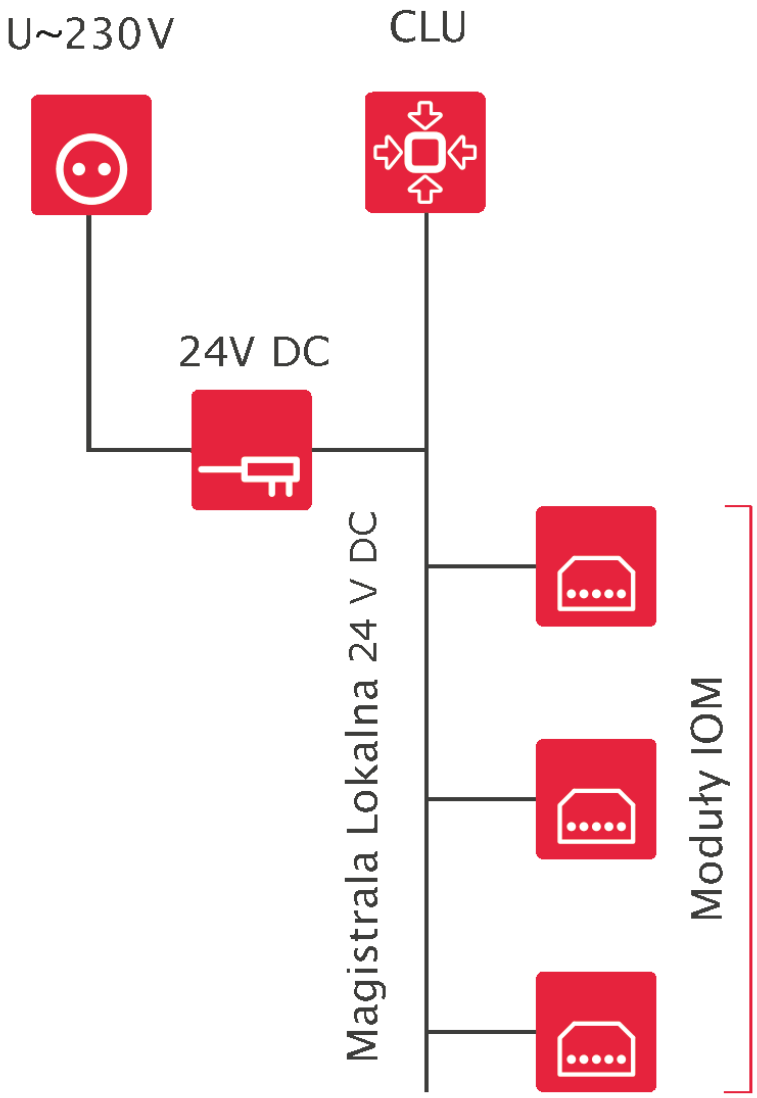
3. Zasilanie modułów

Zasilanie modułów CLU oraz IOM może być zrealizowane na dwa sposoby:

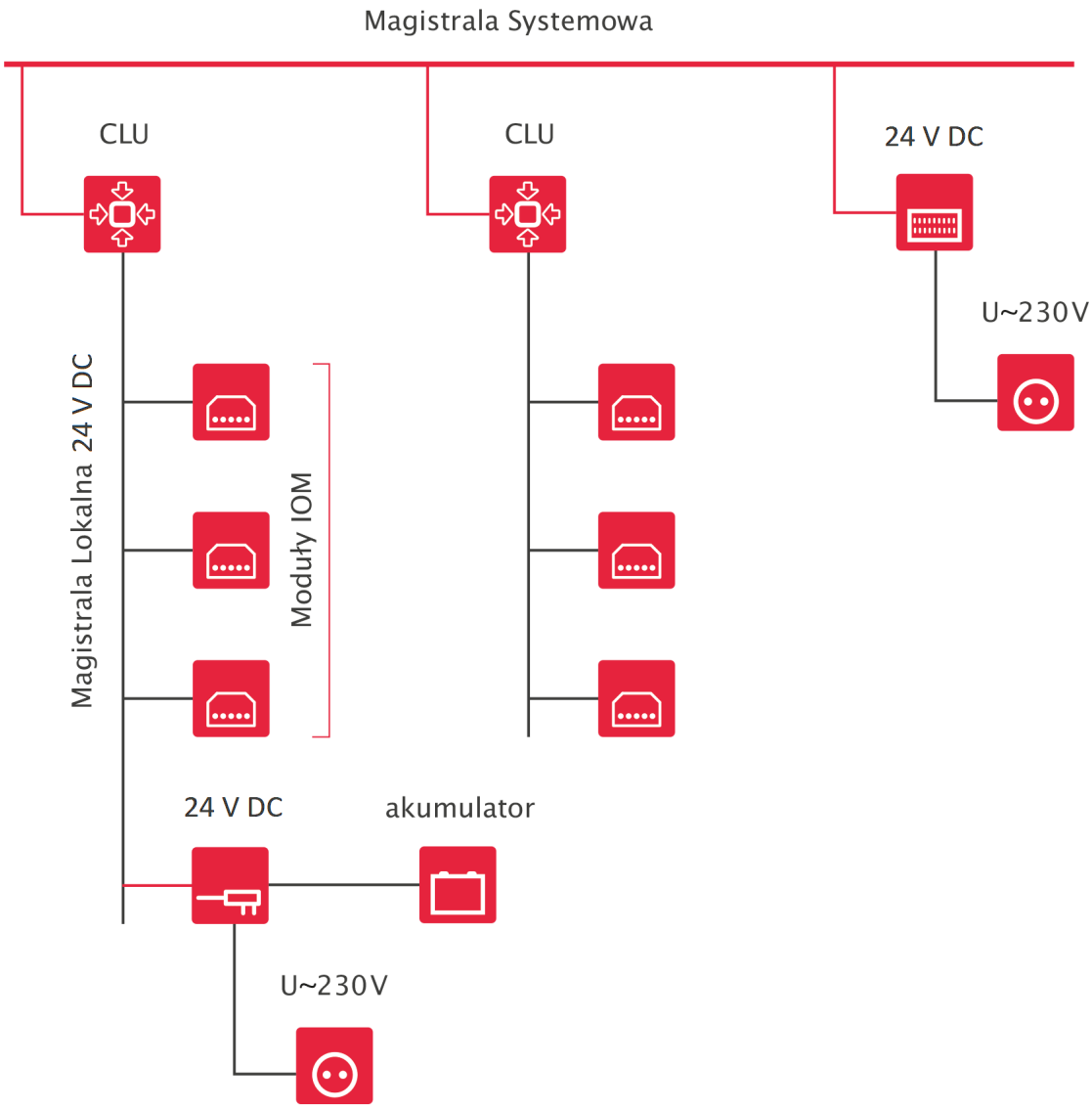
1. Poprzez podłączenie zasilania do magistrali systemowej 24V DC - w takiej sytuacji moduł CLU będzie zasilał podłączone do niego moduły IOM po magistrali lokalnej. Przez magistralę lokalną (TFbus) może przepływać prąd o maksymalnym natężeniu 1000mA.



2. Poprzez podłączenie zasilania 24 V DC do magistrali lokalnej. W takiej sytuacji CLU będzie zasilane z magistrali lokalnej.



W przypadku modułów podtynkowych możliwe jest opcjonalne wykorzystanie zasilacza podtynkowego 24 V DC.



Uwaga! CLU może być jednocześnie podłączone do zasilania z magistrali systemowej oraz magistrali lokalnej!

IV. Instalacja komponentów

Większość modułów dostarczanych jest w dwóch rodzajach wykonania: na szynę DIN (do montażu w rozdzielni) i w wersji podtynkowej. Dodatkowo dostępne są moduły Z-Wave: Relay, Roller Shutter oraz Digital IN.

1. Montaż modułów na rozdzielni

Moduły oferowane przez GRENTON dostarczane są w obudowach przystosowanych do montażu w rozdzielniach na szynie DIN. Aby zamontować moduł, należy założyć moduł na szynę oraz zablokować zatrzask znajdujący się od dołu modułu. Następnie podłączyć moduły do magistrali systemowej za pomocą specjalnych złącz magistralnych oraz podpiąć przewody przyłączeniowe zgodnie z dołączoną do modułów instrukcją instalacji.

> **Uwaga!** Moduły w OM identyfikowane są za pomocą nr seryjnego. Po zainstalowaniu modułu należy zapisać jego nr seryjny oraz fizycznie podpięte wejścia/wyjścia - ułatwi to identyfikację konkretnych obiektów.

2. Montaż modułów podtynkowych kablowych

Moduły przeznaczone do montażu podtynkowego przystosowane są do montażu w puszkach o średnicy 70 mm oraz większości puszek 60 mm. W przypadku modułów podtynkowych zaleca się stosowanie puszek z kieszenią boczną. Dla puszek 60 mm należy dokonać sprawdzenia, czy moduły mieszczą się w tym konkretnym typie puszki.

Do montażu większej liczby modułów, należy stosować puszki pogłębiane.

3. Montaż modułów podtynkowych Z-Wave

Moduły bezprzewodowe przystosowane są do montażu w puszkach instalacyjnych o średnicy minimum 60 mm. Dla modułów podtynkowych zaleca się stosowanie puszek z kieszenią boczną.

V. Object Manager

1. Instalacja OM

Minimalne wymagania sprzętowe dla komputera oraz szczegółowa instrukcja instalacji programu konfiguracyjnego Object Manager dołączona jest do plików instalacyjnych oprogramowania.

Aktualną wersję Object Managera można pobrać ze strony: https://www.grenton.pl/wsparcie/materialy-do-pobrania.html.

> **Uwaga!** Folder, w którym będzie instalowany Object Manager nie może zawierać w nazwie znaków specjalnych tj. %, !, # itd.

A. Windows

- Pobrać plik .exe
- Uruchomić pobrany plik.
- Wybrać ścieżkę instalacji Object Managera.
- Rozpocząć wypakowywanie klikając w przycisk Extract.
- Po wypakowaniu uruchomić plik om.exe znajdujący się w katalogu /object-manager

B. macOS

> **Uwaga!** Nazwa aplikacji zawiera numer wersji co umożliwia istnienie wielu wersji Object Managera na jednym komputerze. Przed usunięciem poprzednich wersji zalecane jest wykonanie migracji/kopii plików projektów domyślnie przechowywanych w pliku aplikacji. W tym celu należy w programie Finder wybrać opcję (Ctrl-Click) "Pokaż Zawartość Pakietu" (Show Package Contents) i skopiować lub przenieść folder zawierający pliki projektu (om.app/Contents/MacOS/projects) do nowej wersji aplikacji Object Managera.

- Pobrać plik
- Uruchomić pobrany plik
- Skopiować aplikację Object Managera do folderu Aplikacji zgodnie z sugestią
- Uruchomić aplikację Object Managera w standardowy sposób

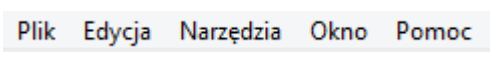
C. Linux

- Pobrać plik object-manager.tar.gz
- Wypakować pobrany plik w wybranej lokalizacji
- Po wypakowaniu uruchomić plik ./om znajdujący się w katalogu /object-manager

2. Struktura OM

Obsługa Object Manager'a dokonywana jest za pomocą trzech dostępnych dla użytkownika paneli menu:

- **główne**



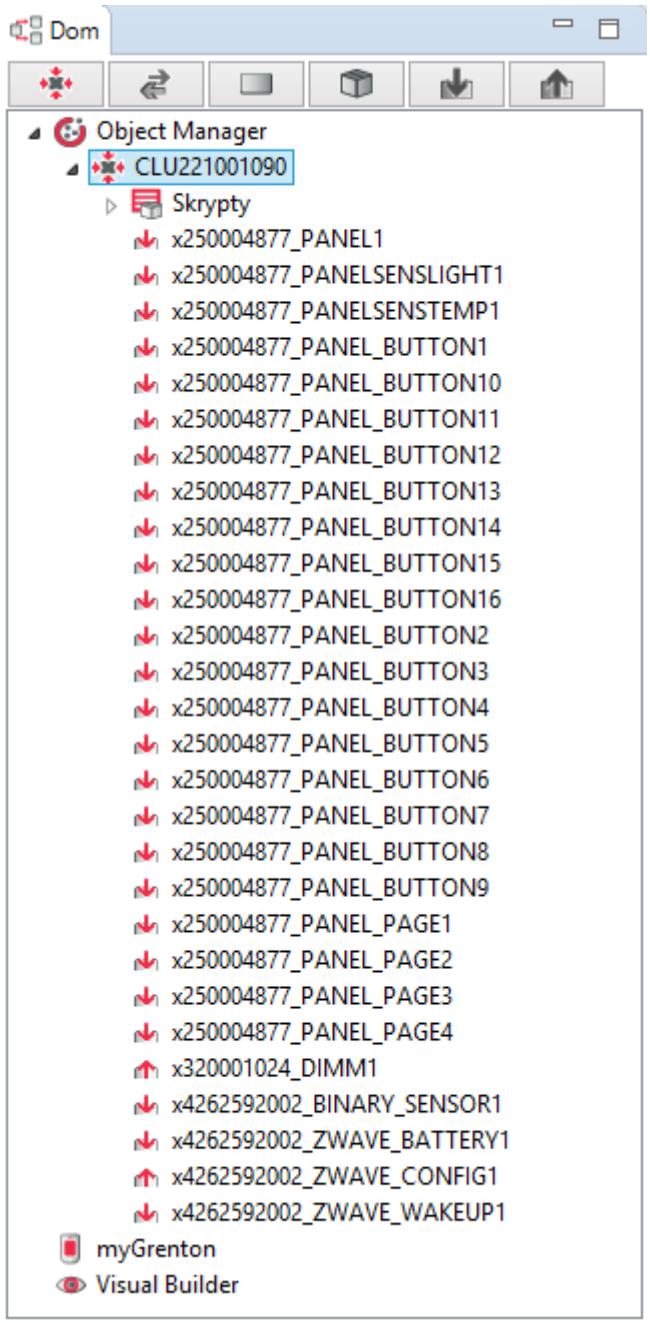
Zawiera podstawowe komendy służące do obsługi projektu.

- **akcji**



Ikony zawarte w tym menu wykorzystywane są przy programowaniu i konfigurowaniu urządzeń. Podświetlone są tylko te ikony, które w danej chwili mogą zostać użyte. Wynika to z kontekstu, w którym w danej chwili się znaleziono (np. jeżeli w bocznym drzewku wybrano CLU, to aktywne staja się ikony związane z CLU).

- **obiektów**



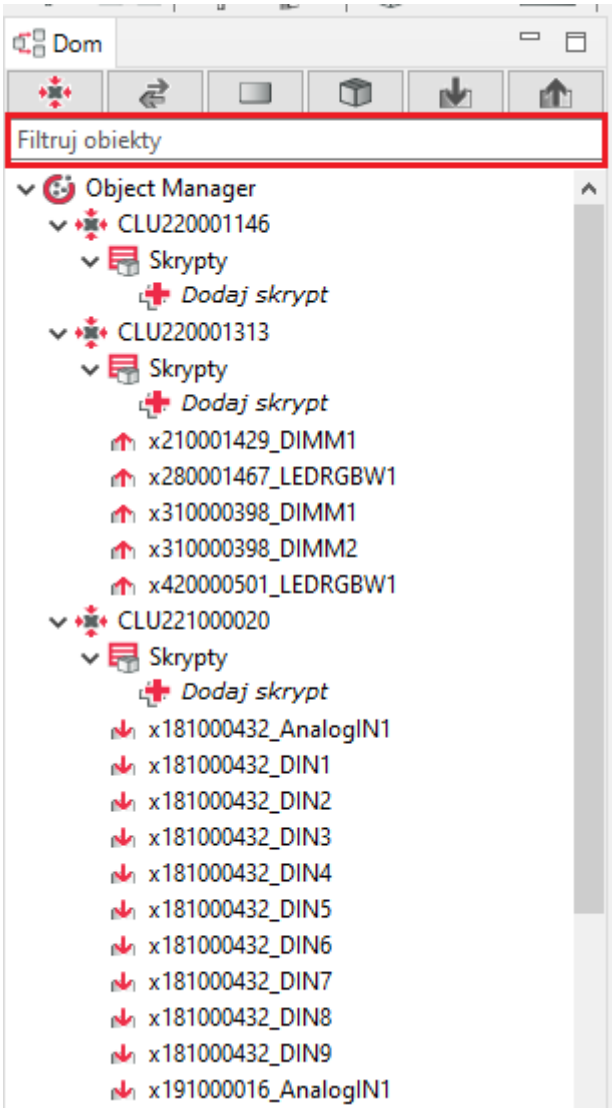
Składa się z trzech części: listy obiektów (CLU, wejść, wyjść), zakładki myGrenton (tworzenie interfejsu aplikacji myGrenton) oraz Visual Buildera (tworzenie interfejsu aplikacji Home Manager).

Wszystkie dane o konfiguracji systemu przechowywane są w pliku projektu. W OM można przechowywać dowolną liczbę projektów, z których każdy jest związany z inną instalacją/budynkiem/apartamentem.

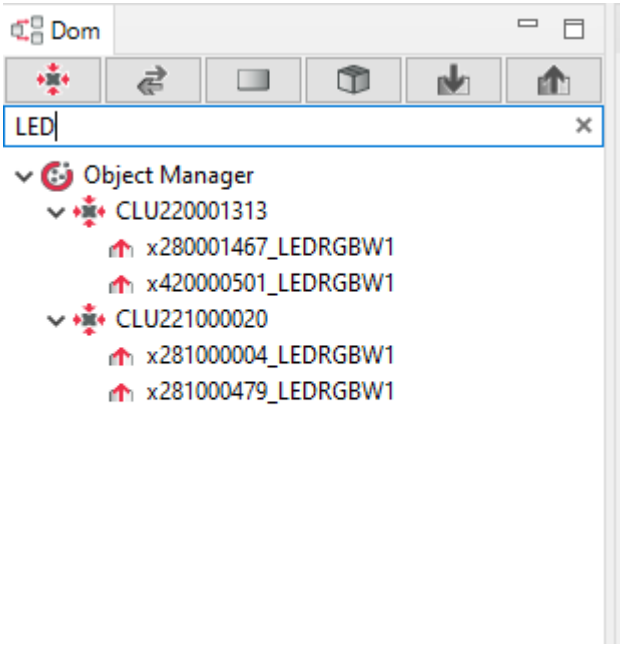
2.1. Filtrowanie obiektów

> **Uwaga!** Funkcjonalność `Filtruj obiekty` dostępna jest dla Object Manager w wersji 1.5.0 lub wyższej.

Dla każdego widoku grupowania (wg CLU, wg modułu, wg typu, wg kontenerów, tylko wejścia, tylko wyjścia) możliwe jest filtrowanie wyświetlanych elementów za pomocą opcji `Filtruj obiekty`.



Po wprowadzeniu wyszukiwanej nazwy, wyświetlane są wszystkie elementy (obiekty) zawierające wpisaną frazę w danym widoku.



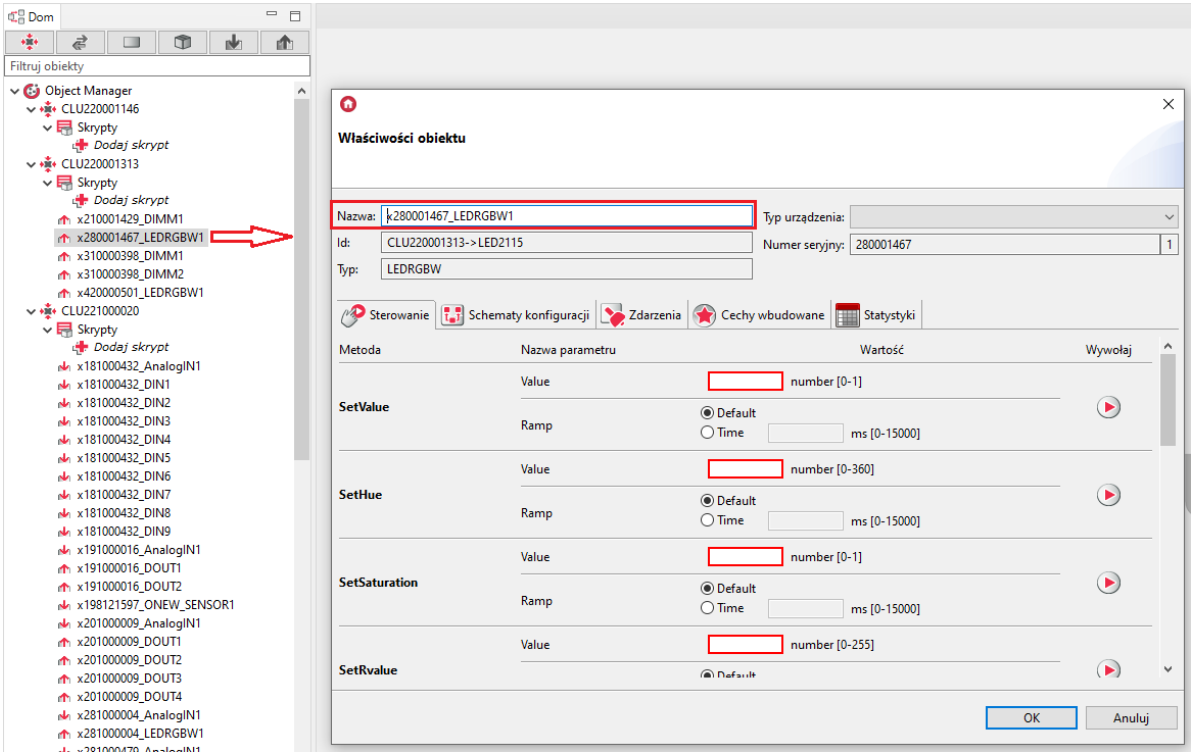
Podana fraza w filtrowaniu jest uwzględniana po przejściu do innego widoku (zakładki) grupowania.

2.2. Zmiana nazwy obiektu

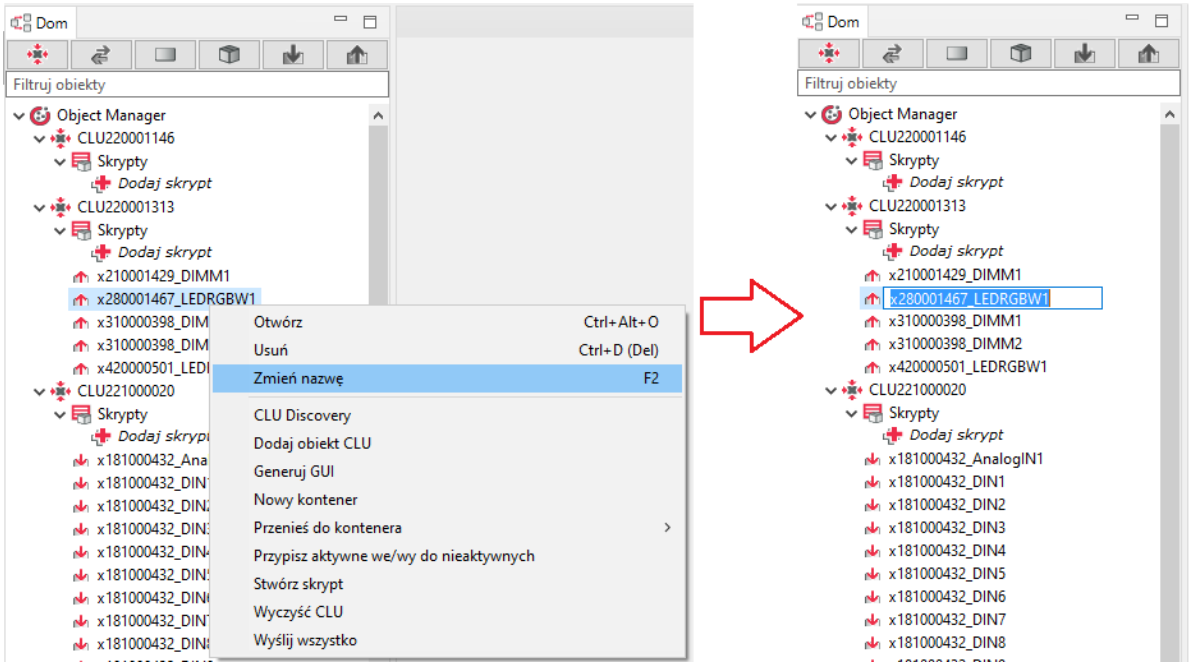
> **Uwaga!** Możliwość zmiany nazwy za pomocą menu kontekstowego / `F2` dostępna jest dla Object Manager w wersji 1.5.0 lub wyższej.

Zmianę nazwy danego obiektu można wykonać w następujący sposób:

- w oknie Właściwości obiektu wprowadzenie nowej nazwy w polu `Nazwa`,



- zaznaczenie obiektu i wywołanie opcji zmiany nazwy z menu kontekstowego lub za pomocą skrótu klawiszowego `F2`.



> **Uwaga!** Zmiana nazwy z poziomu menu kontekstowego / `F2` nie jest dostępna dla elementów interfejsu Home Manager.

3. Pliki projektów

3.1. Katalog zapisanych projektów

Po zainstalowaniu Object Managera należy wskazać katalog, w którym będą przechowywane zapisane projekty.

Domyślna ścieżka dostępu do katalogu: `C :\.... \OM\projects`

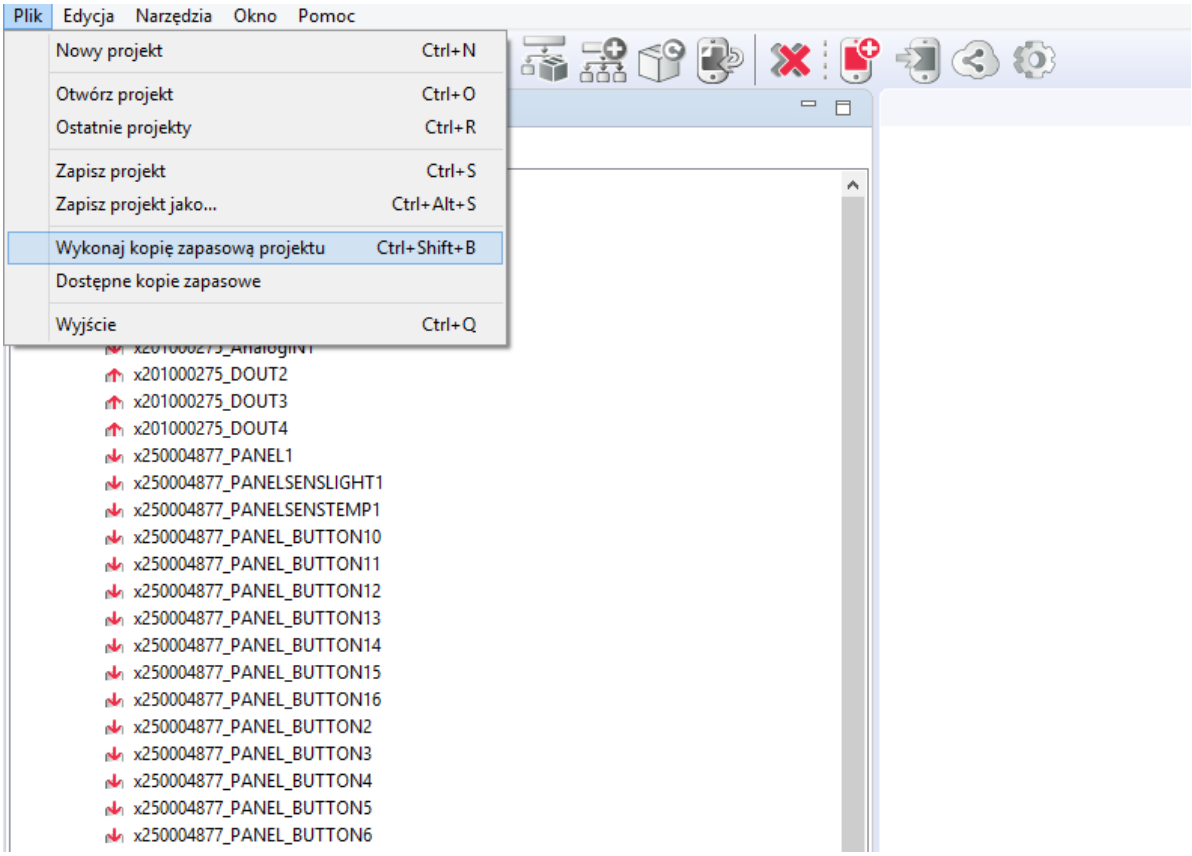
Wszystkie pliki utworzonych i zapisanych projektów zapisywane są w tym katalogu z rozszerzeniem `*.omp` (np. `projekt.omp`).

3.2. Kopia zapasowa projektu (backup)

W trakcie pracy nad projektem, istnieje możliwość wykonania kopii zapasowej projektu, która nie ulega modyfikacji, mimo wykonania zmian w projekcie. Dzięki temu istnieje możliwość odzyskania wcześniejszej wersji projektu, gdyby użytkownik wykonał niepożądane zmiany w konfiguracji. Do każdego projektu można utworzyć dowolną liczbę kopii zapasowych.

> **Uwaga!** Zaleca się możliwe najczęstsze wykonywanie kopii zapasowych projektu - zwłaszcza przed wykonaniem istotnych zmian w konfiguracji systemu.

By utworzyć kopię zapasową projektu, należy z menu głównego wybrać pozycję `Plik->Wykonaj kopię zapasową projektu` (kopię można również wykonać skrótem klawiszowym `CTRL+Shift+B`).



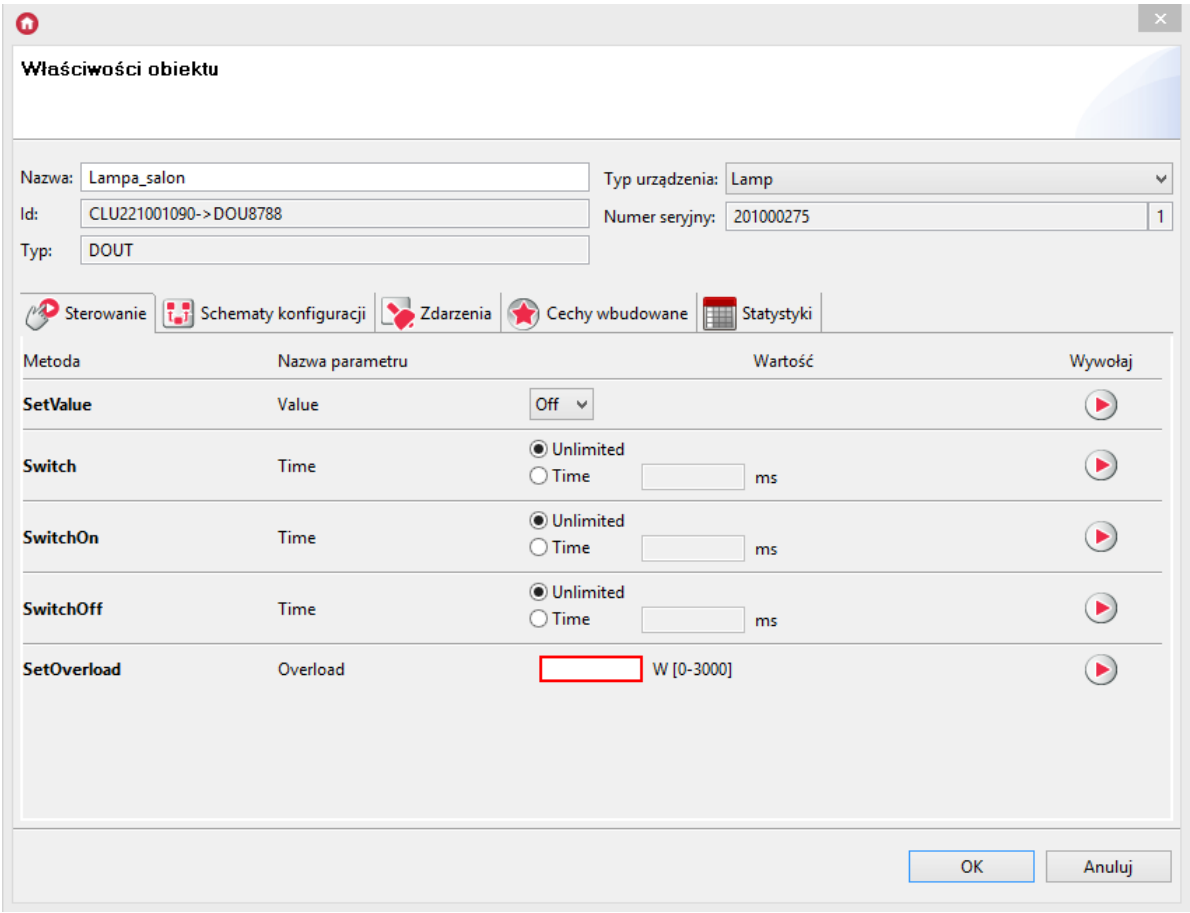
Zapisane kopie dostępne są na liście otwieranej po kliknięciu w `Dostępne kopie zapasowe` lub w oknie otwierania projektu w zakładce `Kopie zapasowe`.

> **Uwaga!** Po wybraniu kopii zapasowej z listy zostanie ona załadowana, a zmiany obecnie wprowadzone w projekcie, jeśli nie były zapisane, zostaną utracone.

4. Podstawowe elementy

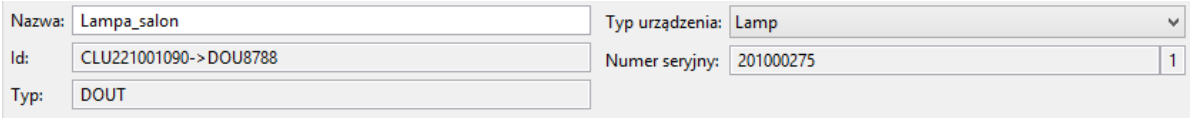
4.1. Konfigurator obiektów

Każde wejście, wyjście, sensor czy inne urządzenie fizycznie podpięte do systemu ma swoje odzwierciedlenie w OM w postaci obiektów. Obiekty nie odzwierciedlają fizycznych modułów, ale poszczególne wejścia i wyjścia. Każdy obiekt posiada swoje cechy wbudowane wraz z ich wartościami początkowymi oraz zdarzenia, które potrafi wygenerować. Okno konfiguratora wyświetli się po podwójnym kliknięciu w dany obiekt.



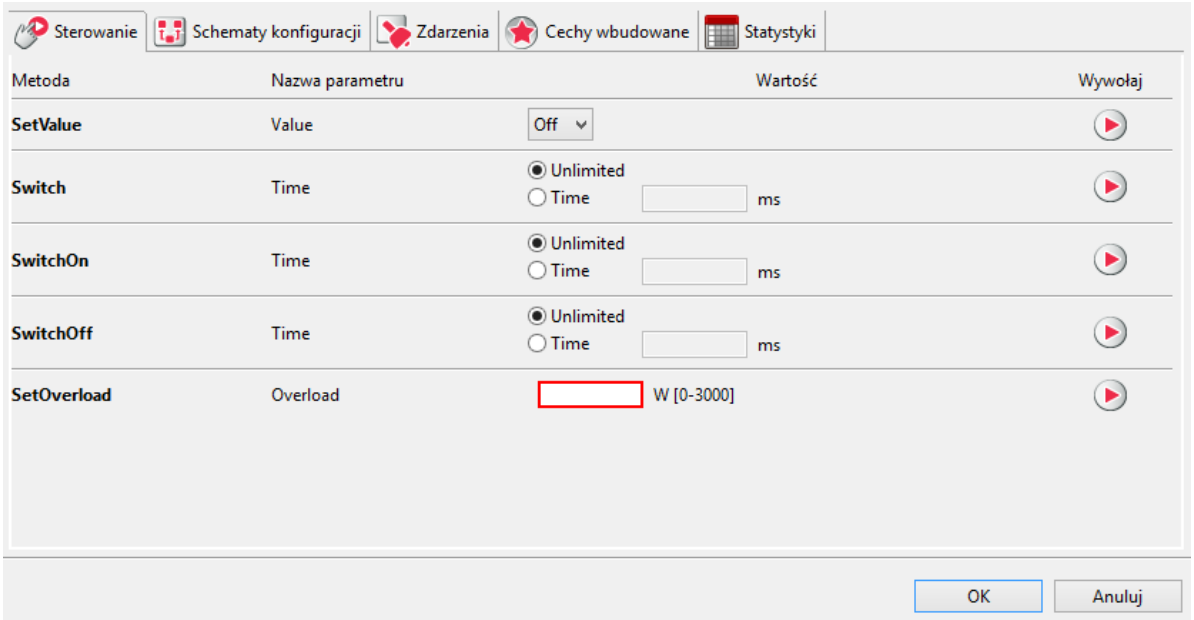
Powyższa formatka zawiera następujące sekcje:

1. Informacje podstawowe



Sekcja mieści się w górnej części formatki i zawiera podstawowe informacje właściwe dla każdego obiektu, np.: nazwę, Id modułu, jego typ, numer seryjny oraz numer wejścia/wyjścia w ramach danego modułu. W polu *Typ urządzenia* użytkownik może zdefiniować rodzaj źródła lub odbiornika fizycznie podłączonego do tego obiektu.

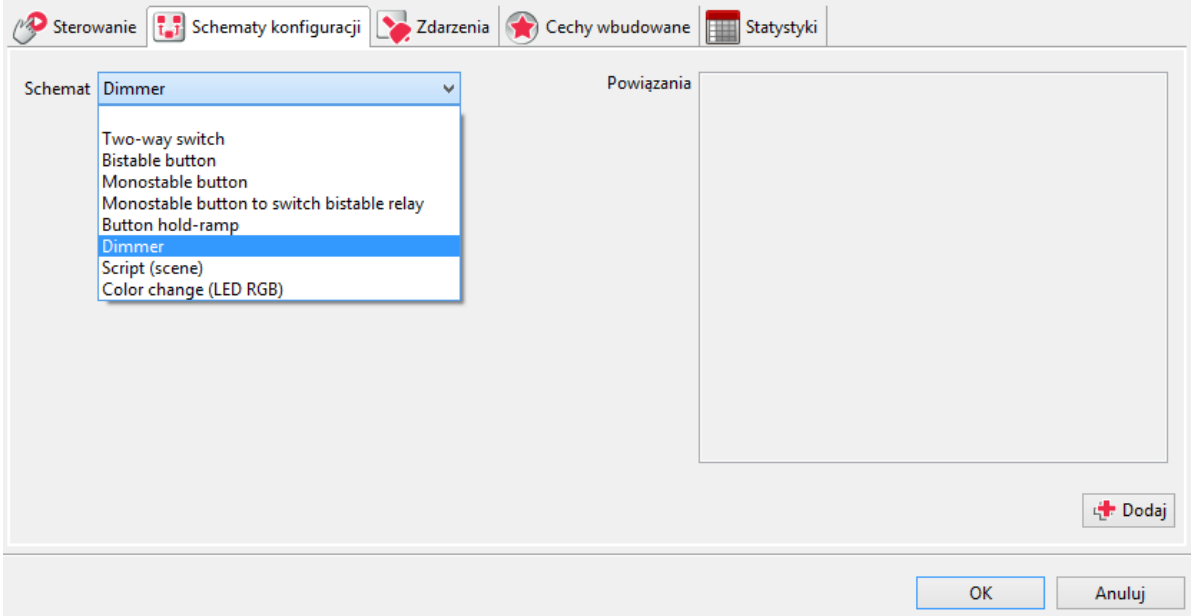
2. Sterowanie



Zawiera metody (wraz ze wszystkimi parametrami), właściwe dla przeglądanego obiektu. Umożliwia wywołanie danej metody z poziomu OM. Przykładowo dla wyjścia przekaźnikowego możemy wywołać metodę `SwitchOn` z parametrem *Time* (np. 30 s), co spowoduje załączenie tego wyjścia na czas 30 s. W celu wywołania metody z poziomu OM należy - w zakładce *Sterowanie* - wprowadzić wartości parametrów wywoływanej metody (o ile są konieczne) i wcisnąć przycisk *Wywołaj*.

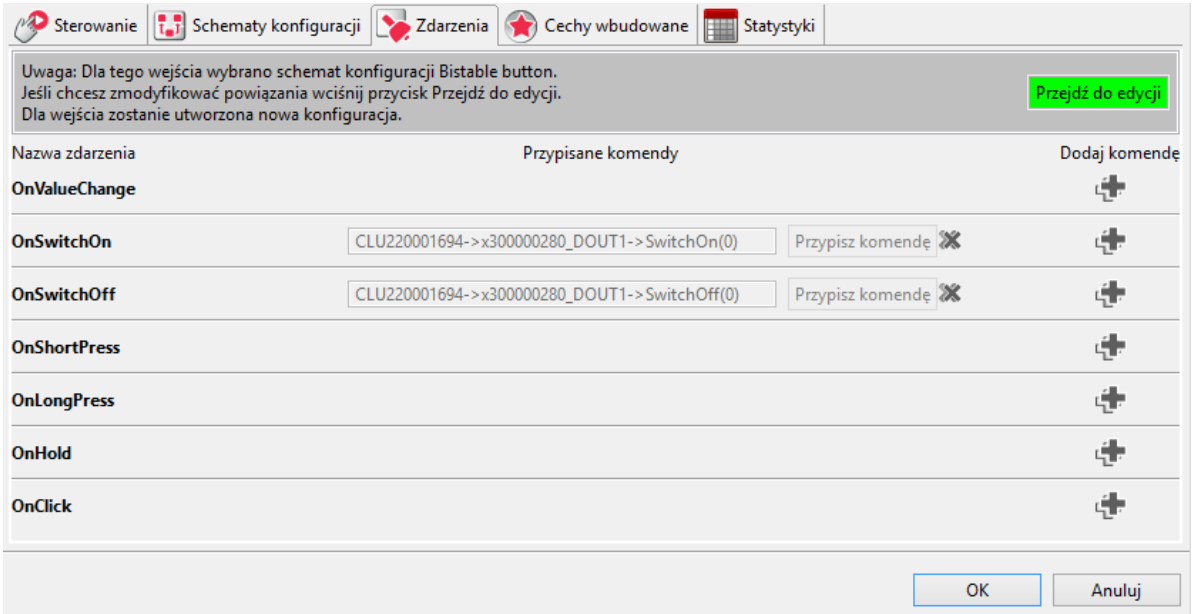
3. Schematy konfiguracji

Schematy konfiguracji definiują sposób zachowania obiektu i pozwalają na uproszczoną konfigurację logiki. Aby dodać schemat konfiguracji należy wybrać dostępny schemat z pola wyboru, a następnie dodać powiązanie z modułem klikając Dodaj - patrz pkt VI.9..



4. Zdarzenia

Zakładka ta zawiera listę zdarzeń odpowiednią dla danego typu obiektu oraz przypisanych im metod, które są wywoływane po zaistnieniu zdarzenia (jeśli są zdefiniowane). W przypadku, gdy wybrany został schemat konfiguracji, zakładka jest w trybie tylko do odczytu i pokazuje jedynie powiązania stworzone w ramach wybranego schematu.



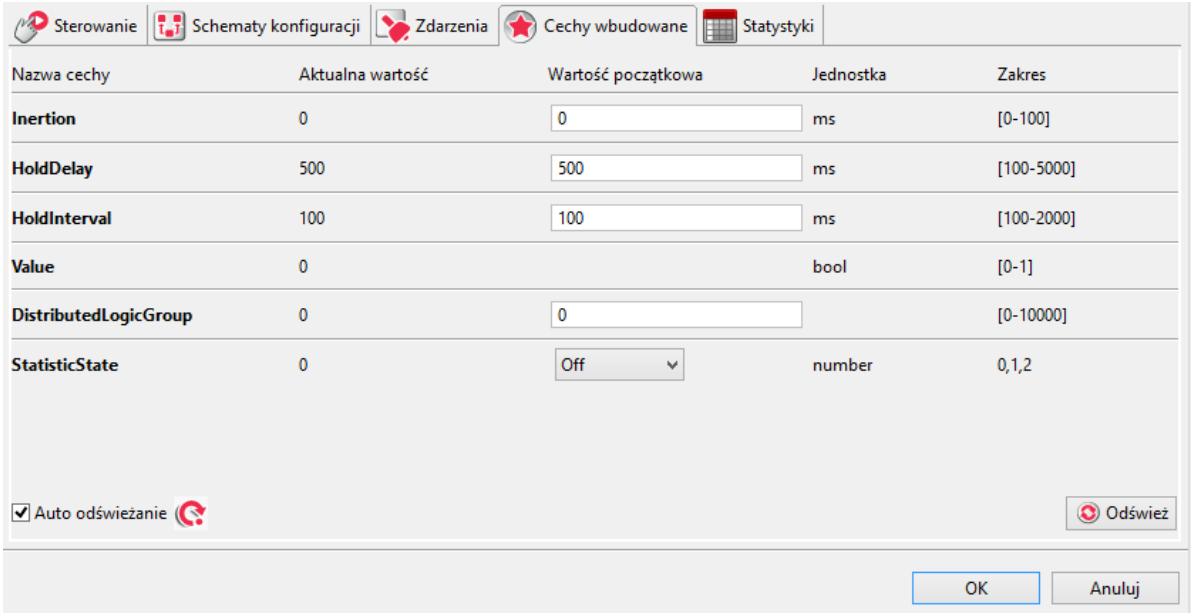
W każdej chwili można przejść do edycji powiązań zdarzenie-metoda, klikając *Przejdź do edycji*. W takiej sytuacji utworzony zostanie schemat *Konfiguracja użytkownika*, który pojawi się na liście w zakładce Schematy konfiguracji.



Po wciśnięciu przycisku *Dodaj komendę* do wybranego zdarzenia, otwiera się lista obiektów. Następnie po zaznaczeniu interesującego nas obiektu pojawia się lista metod, jakie można na nim wywołać. Dodanie wybranej metody powoduje utworzenie nowej zależności pomiędzy obiektami.

5. Cechy wbudowane

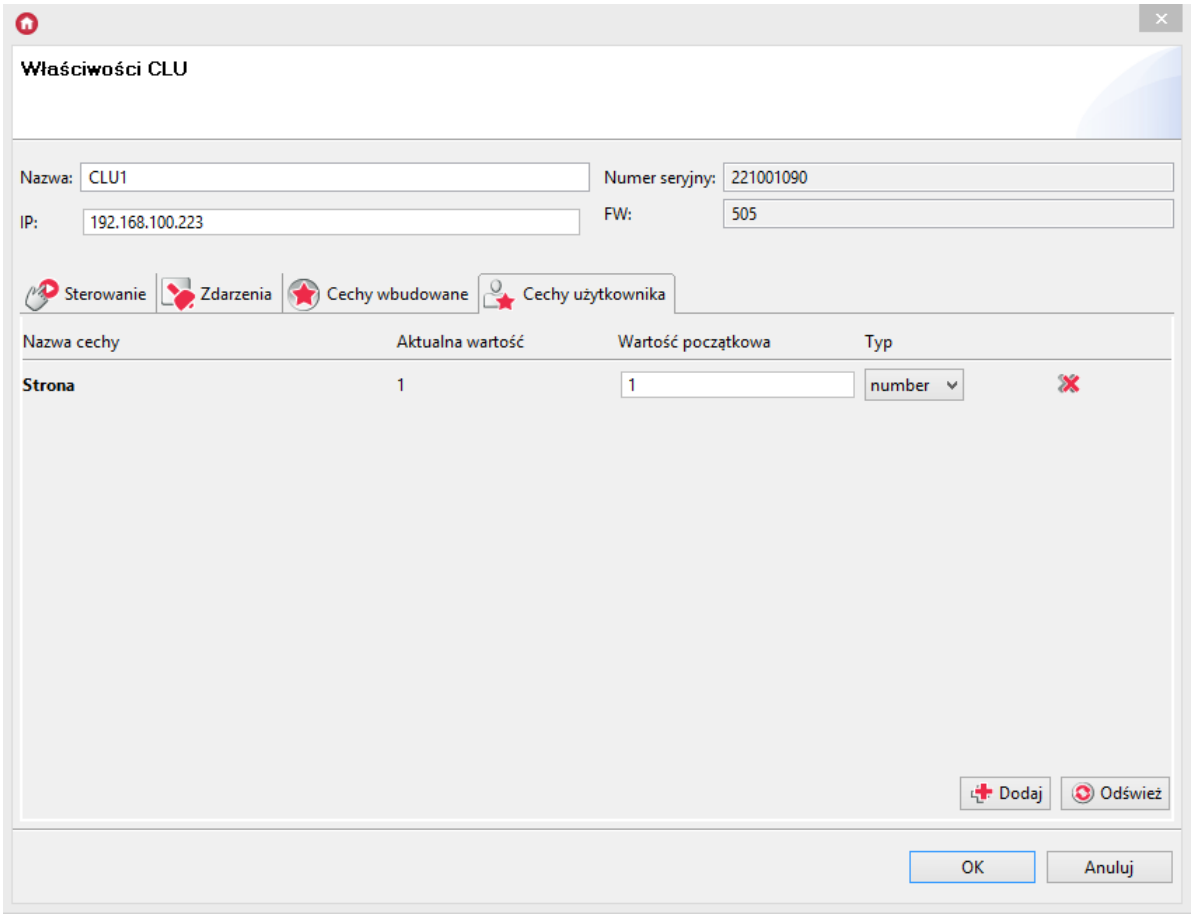
W tej części przedstawione są wartości, jakie aktualnie posiada wybrany obiekt oraz wartości początkowe, które zostały w nim zapisane (wartości początkowe ustawiane są w przypadku restartu systemu, np. po zaniku zasilania). Wpisanie wartości w polu *Wartości początkowe* spowoduje ustawienie jej podczas startu CLU.



6. Cechy użytkownika (tylko CLU)

Zakładka ta pozwala użytkownikowi zdefiniować na CLU własną listę cech, które mogą być następnie używane do przechowywanie rożnego typu danych (liczniki, znaczniki). Dodanie cechy użytkownika następuje po kliknięciu przycisku 

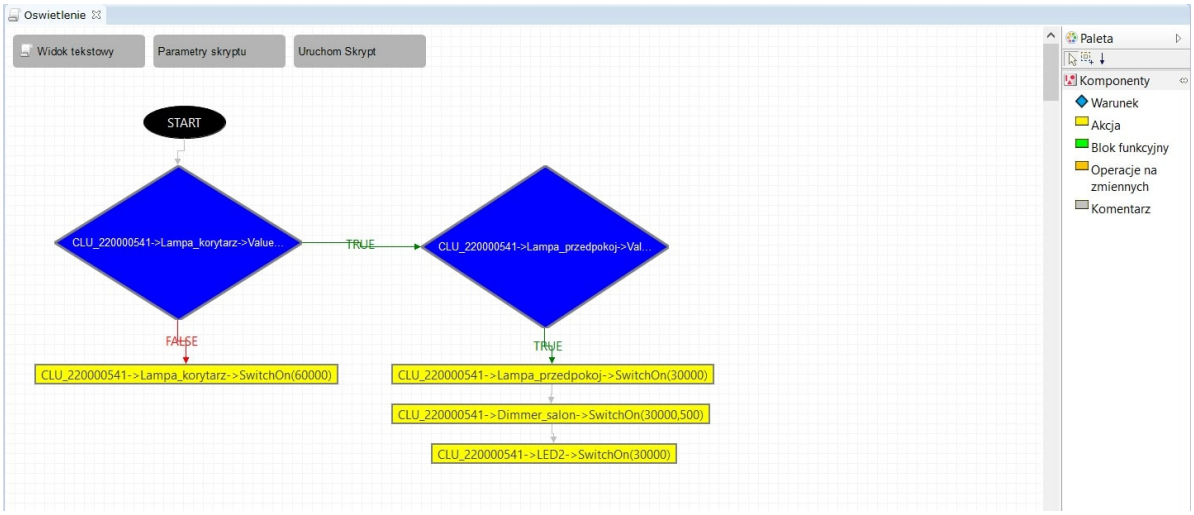
i podaniu nazwy cechy. Następnie należy zdefiniować wartość początkową oraz typ cechy (tekstowa, numeryczna lub boolowska).



4.2. Script builder

To narzędzie służące do tworzenia skryptów, które może działać w dwóch trybach:

1. **Graficznym** (uproszczonym), w którym w prosty sposób możemy stworzyć schemat, poprzez przeciąganie oraz łączenie elementów.



Tryb graficzny umożliwia tworzenie skomplikowanych skryptów składających się z wielu warunków oraz metod. Możliwe jest również wykorzystanie zmiennych i parametrów. Jedynym ograniczeniem jest brak możliwości tworzenia pętli, które wymagają użycia trybu tekstowego.

2. **Tekstowym** (pełnym), w którym użytkownik ma możliwość tworzenia logiki z wykorzystaniem rozszerzonego języka LUA. Dzięki temu możliwe jest tworzenie bardzo zaawansowanych skryptów z wykorzystaniem wszelkich elementów języka LUA (w tym pętli, tablic itp.).



W stosunku do standardowego języka LUA, język został rozszerzony o możliwość bezpośredniego odwoływania się do adresów, metod i cech, które traktowane są tak samo jak inne funkcje LUA.

4.3. Diagram połączeń

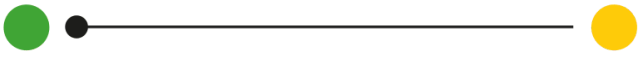
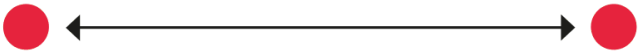
Narzędzie obrazujące zależności oraz powiązania pomiędzy wszystkim obiektami znajdującymi się w systemie. Dzięki niemu można w prosty i czytelny sposób znaleźć interesującą zależność lub bez konieczności przeszukiwania konfiguracji sprawdzić zależności dla danego modułu.

Diagram połączeń można uruchomić z menu górnego: *Narzędzia->Diagram połączeń* lub skrótem klawiszowym [`ALT+Q`].

Każdy obiekt w systemie prezentowany jest na diagramie jako koło z wyświetlonym obok adresem. Kolor koła uzależniony jest od typu obiektu:

- CLU - kolor czerwony;
- Wejście/wyjście - kolor wiśniowy;
- Zdarzenia wejść lub wyjść - kolor jasnoniebieski;
- Zdarzenia generowane przez timery - kolor ciemnoniebieski;
- Metody wbudowane - kolor ciemnozielony;
- Metody skryptowe - kolor jasnozielony;
- Cechy wbudowane - kolor żółty;
- Cechy zdefiniowane - kolor pomarańczowy;

Powiązania między obiektami wyświetlane są w postaci strzałek, których grot wskazuje obiekt wyzwalany.

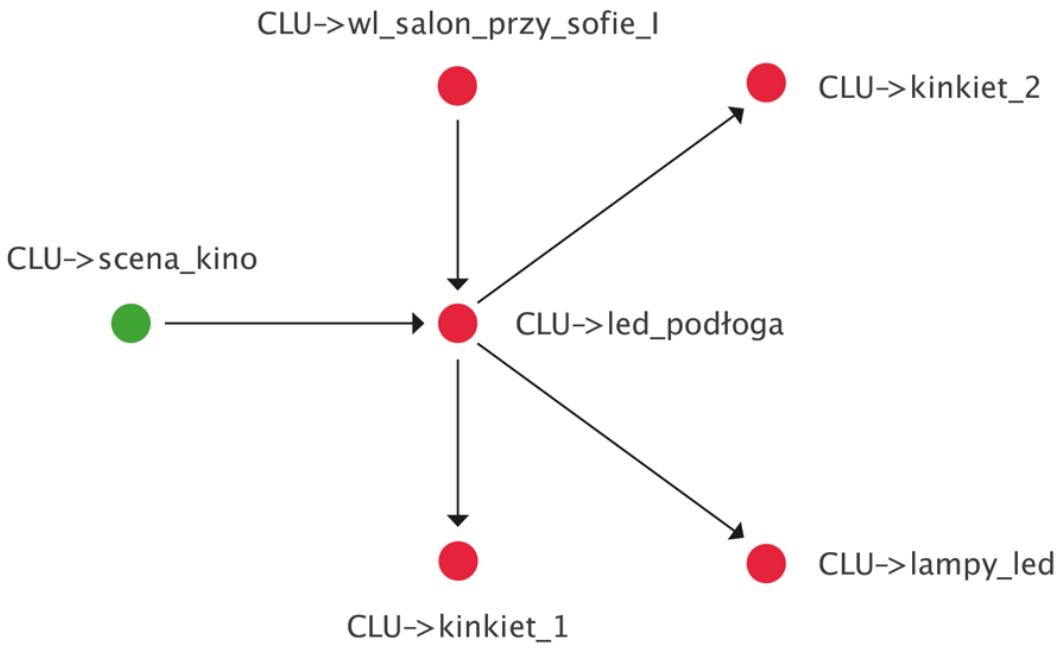
RODZAJ POWIĄZANIA	ZAKOŃCZENIE OD STRONY OBIEKTU AKTYWNEGO	ZAKOŃCZENIE OD STRONY OBIEKTU PASYWNEGO
Bezwarunkowe wywołanie metody		
Wywołanie metody objęte warunkiem		
Odczyt wartości cech		
Zapis wartości cech		
Powiązanie obiektów dwukierunkowe		

Powiązania obrazowane są na trzech poziomach:

1. CLU-CLU – wyświetla powiązania pomiędzy dwoma CLU, jeżeli jakikolwiek obiekt jednego CLU (wejścia/wyjścia) jest powiązany z drugim CLU.
2. Powiązania pomiędzy obiektami – wyświetla powiązania pomiędzy poszczególnymi obiektami (wejściami/wyjściami) bez wskazywania na konkretne zdarzenia, cechy, metody.
3. Powiązania zdarzeń, metod i cech – wyświetla najbardziej szczegółowy widok obrazujący co wywołują konkretne zdarzenia itd.

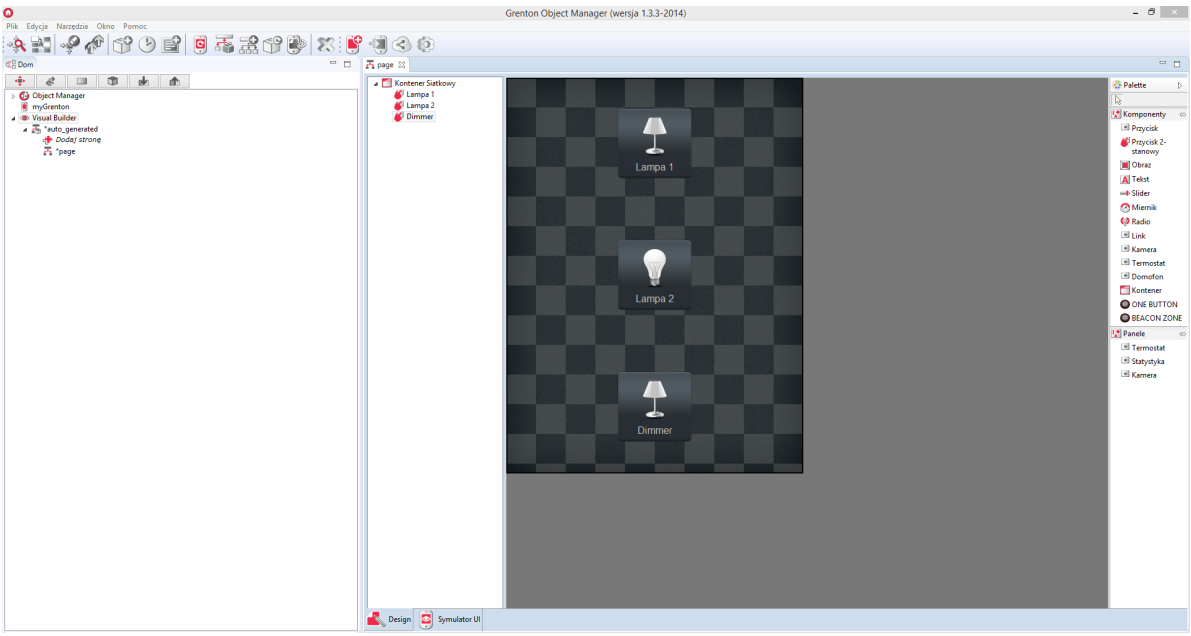
Również nawigacja odbywa się w dwóch płaszczyznach:

1. W płaszczyźnie poziomej – umożliwia przechodzenie pomiędzy obiektami na tym samym poziomie poprzez kliknięcie na dowolny obiekt na diagramie (z wyjątkiem centralnego).
2. W płaszczyźnie pionowej – umożliwia przechodzenie góra-dół, pomiędzy kolejnymi poziomami poprzez kliknięcie na obiekt centralny i wybór obiektu z listy, która się pojawia (przejście w dół) lub poprzez naciśnięcie przycisku „w górę" znajdującego się w górnej części diagramu (przejście w górę).



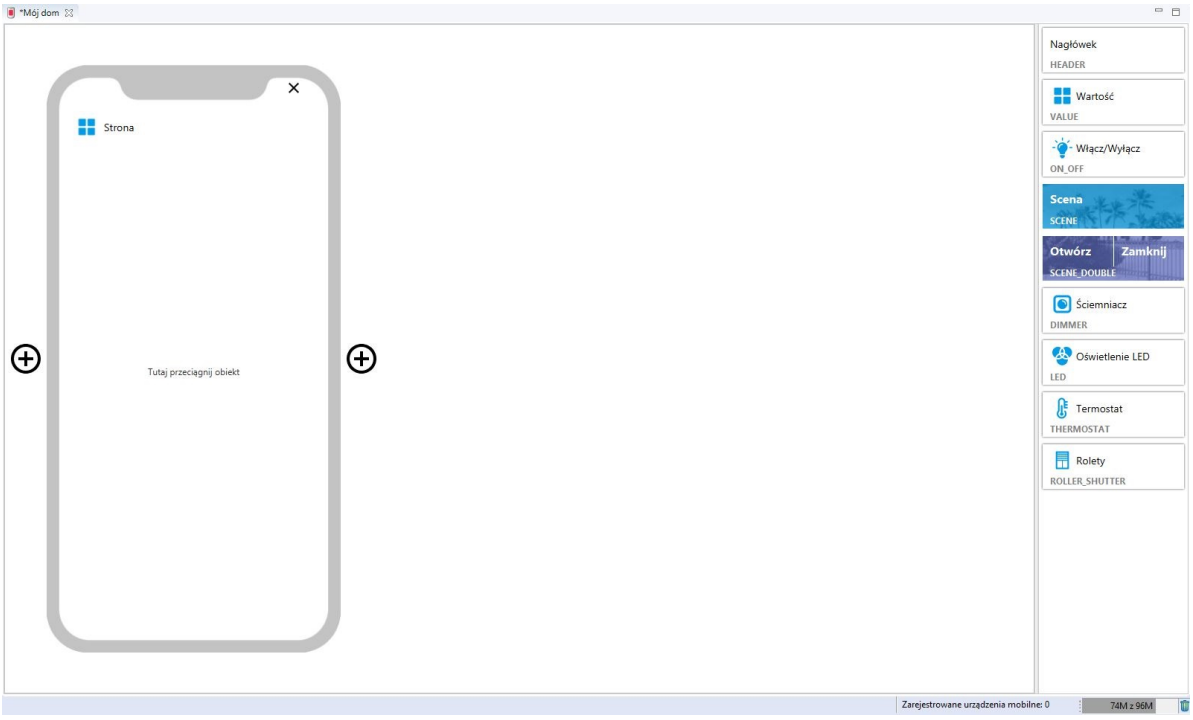
4.4. Visual Builder

Visual Builder jest narzędziem służącym do tworzenia interfejsu użytkownika na urządzenia mobilne dla aplikacji Home Manager. Interfejs może być tworzony automatycznie na podstawie projektu instalacji lub może zostać zaprojektowany i utworzony przez użytkownika według osobistych preferencji. Użytkownik ma możliwość stosowania własnych grafik. Tworzenie interfejsu odbywa się poprzez drag&drop komponentów Visual Buildera i umożliwia utworzenie interfejsu dla wszystkich popularnych rozdzielczości. Ikona uruchamiająca VB znajduje się na końcu rozwijanego drzewka obiektów.



4.5. myGrenton

W Object Managerze znajduje się również narzędzie służące do tworzenia interfejsu użytkownika na urządzenia mobilne dla aplikacji myGrenton. Tworzenie interfejsu odbywa się poprzez drag&drop elementów z drzewa obiektów dostępnych w danym projekcie (modułów podłączonych do CLU) - utworzony widget będzie posiadał uprzednio zdefiniowany szablon.



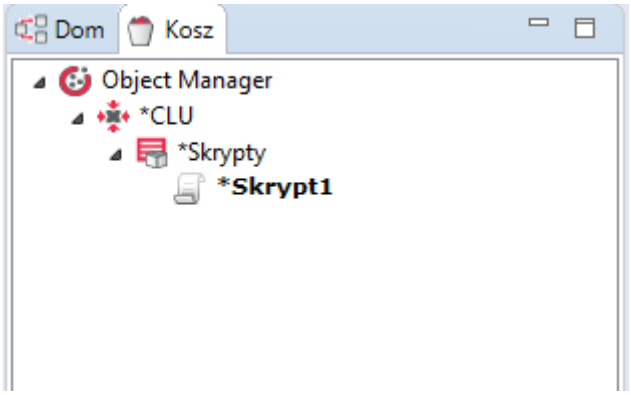
4.6. Kosz

Kosz wzorowany jest na rozwiązaniu znanym z systemów operacyjnych. Usunięty obiekt, skrypt czy aplikacja w projekcie nie jest usuwana bezpowrotnie, ale trafia do kosza dając możliwość użytkownikowi na odzyskanie skasowanych danych w przypadku zmiany koncepcji.

Kosz ma postać zakładki znajdującej się w drzewku obiektów i pojawia się, gdy jakiś obiekt zostanie usunięty. Obiekty z kosza można w dowolnej chwili przywrócić klikając na nie prawym klawiszem i wybierając *Przywróć* z menu kontekstowego.

Obiekt można bezpowrotnie usunąć z kosza wybierając *Usuń* z menu kontekstowego. Przywrócenie do projektu usuniętego w taki sposób modułu jest możliwe jedynie poprzez wykonanie *CLU Discovery*.

Kosz jest świetnym rozwiązaniem do przechowywania obiektów, które nie są w tej chwili wykorzystywane, ale mogą być przydatne w przyszłości.



VI. Podstawowa konfiguracja systemu

1. Połączenie OM z CLU

W celu skonfigurowania urządzenia w systemie, komputer musi być podłączony do modułów CLU. W trakcie przeprowadzania prac, wszystkie moduły CLU muszą być połączone ze sobą przy pomocy kabla Ethernet.

Istnieją dwa sposoby połączenia:

1. Bezpośrednie połączenie z komputerem Kabel sieciowy podłączamy do karty sieciowej w komputerze i łączymy z gniazdem sieciowym w module CLU.
2. Połączenie przez sieć lokalną Można połączyć się z systemem GRENTON wykorzystując sieć lokalną. W tym celu zarówno moduł CLU jak i komputer, z którego będziemy się łączyć, muszą znajdować się w tej samej podsieci.

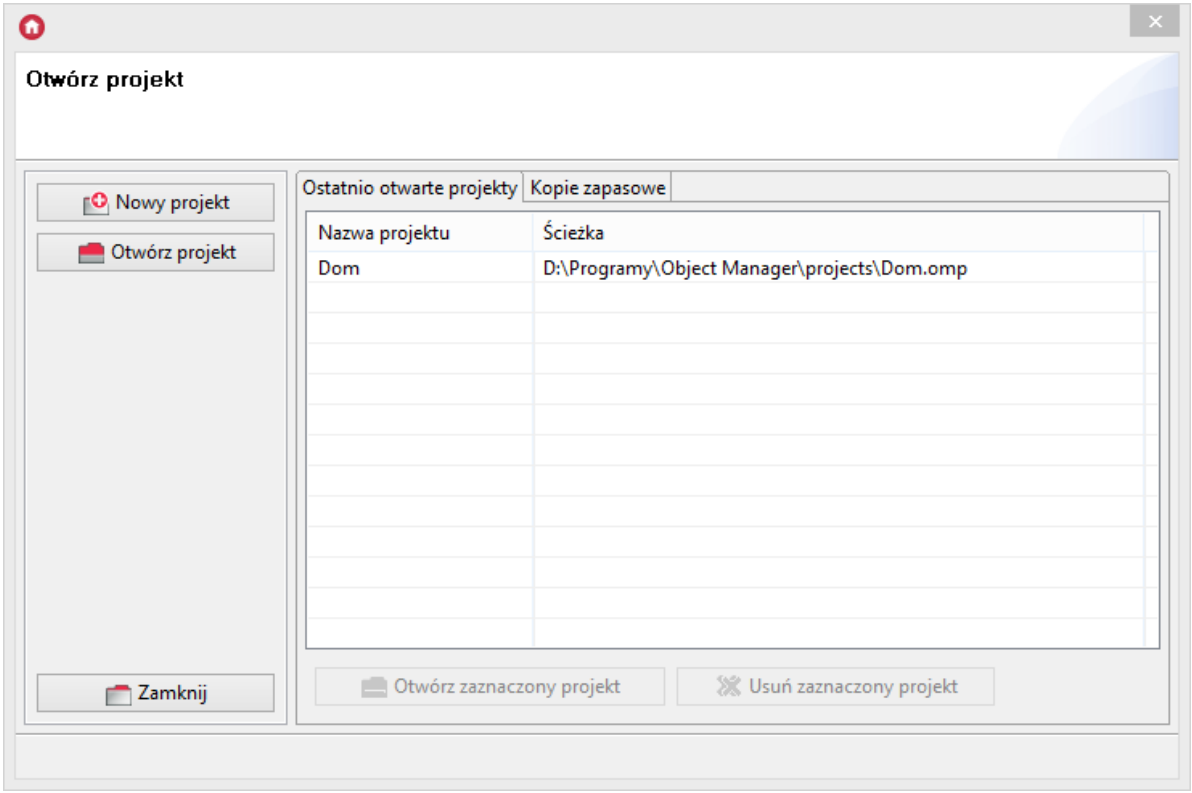
2. Adresy IP

Moduł CLU, jak każde urządzenie sieciowe, otrzymuje swój własny adres IP. Każdy z modułów zainstalowanych w danym systemie musi posiadać swój unikalny adres IP, jednakże wszystkie moduły CLU w systemie muszą pracować w tej samej podsieci, by mogły się ze sobą komunikować. Adres IP dla danego CLU może zostać w dowolnym momencie zmieniony przez użytkownika. Zmiany adresu można dokonać poprzez formatkę konfiguratora urządzeń dla wybranego CLU przez wpisanie w polu zawierającego stary adres, nowego adresu.

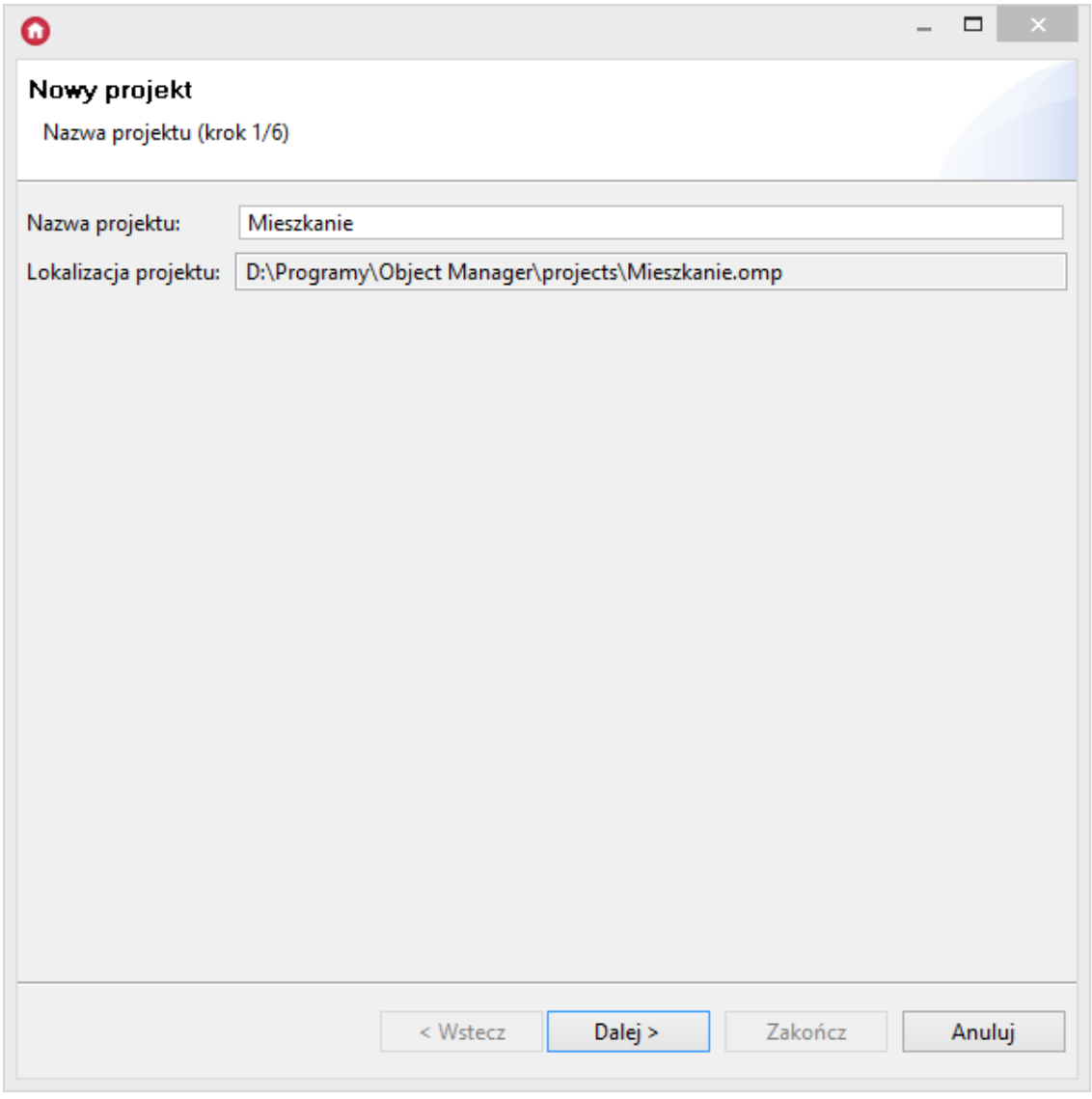
> **Uwaga!** Po podpięciu CLU (lub kilku CLU) do karty sieciowej komputera, zostanie mu nadany nowy adres IP - zgodny z pulą adresów karty sieciowej komputera.

3. Utworzenie nowego projektu

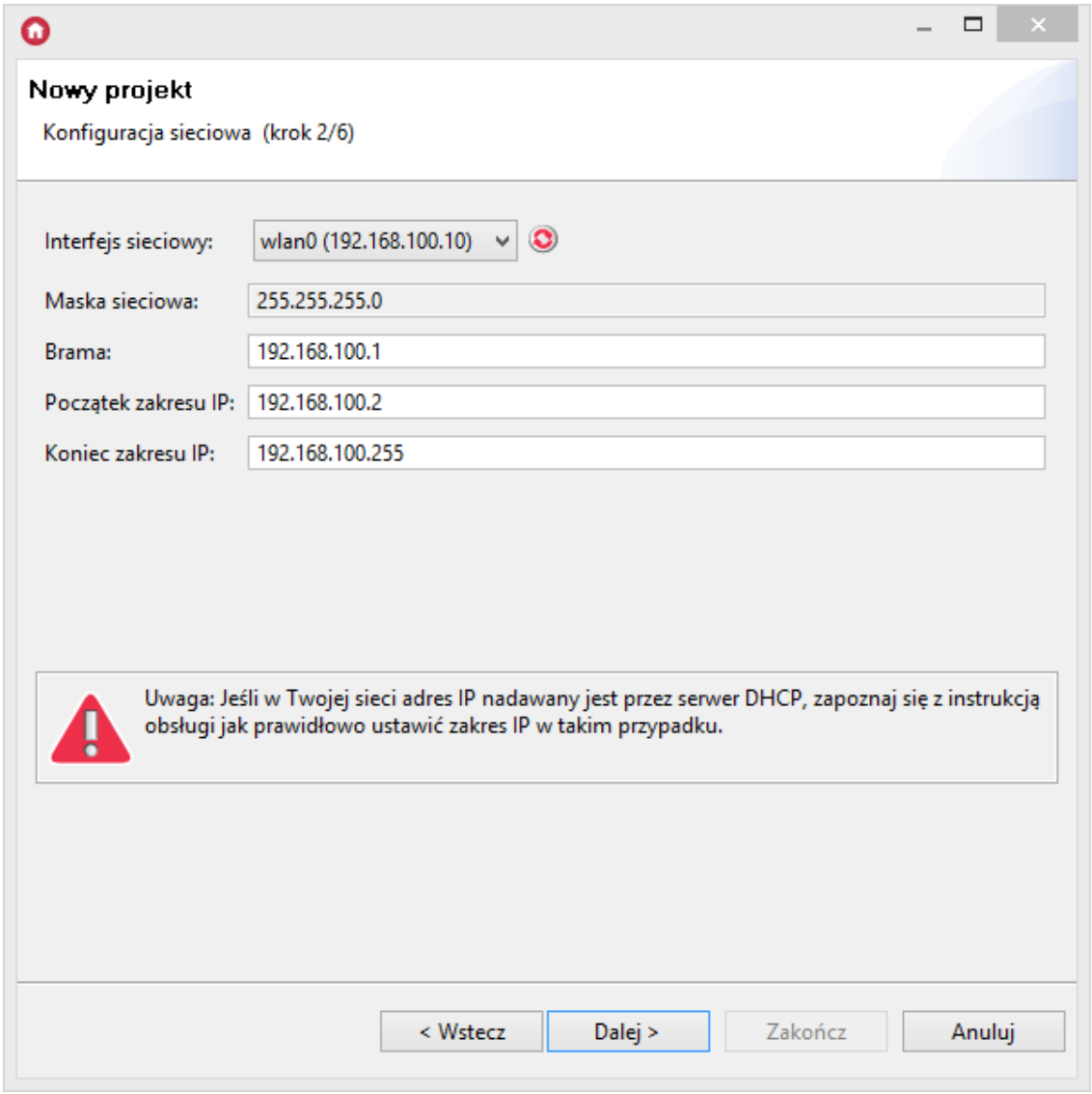
Po otwarciu Object Managera pojawia się okno z możliwością wyboru dwóch opcji: utworzenie nowego projektu oraz otwarcie zapisanego projektu. Dodatkowo mamy opcję wyboru ostatnio otwartych projektów oraz wykonanych kopii zapasowych.



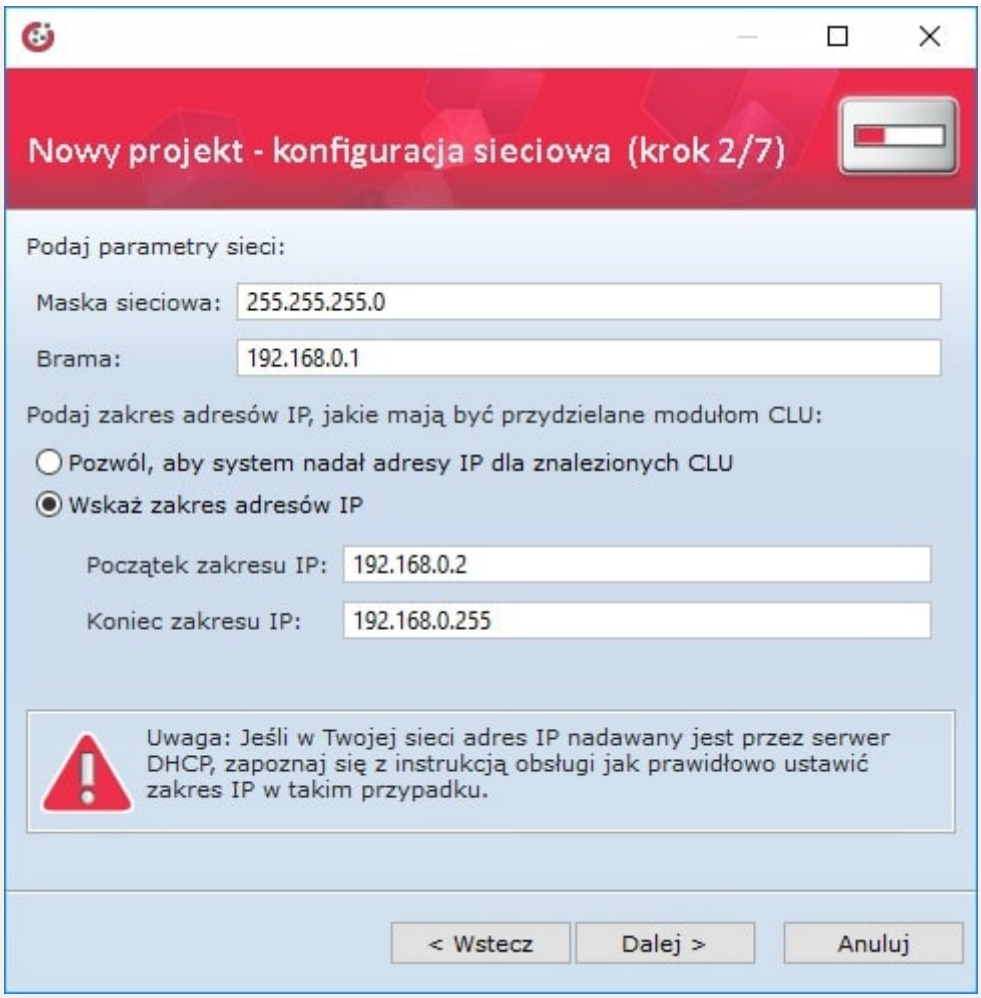
1. Należy wybrać utworzenie nowego projektu, a następnie nadać mu nazwę.



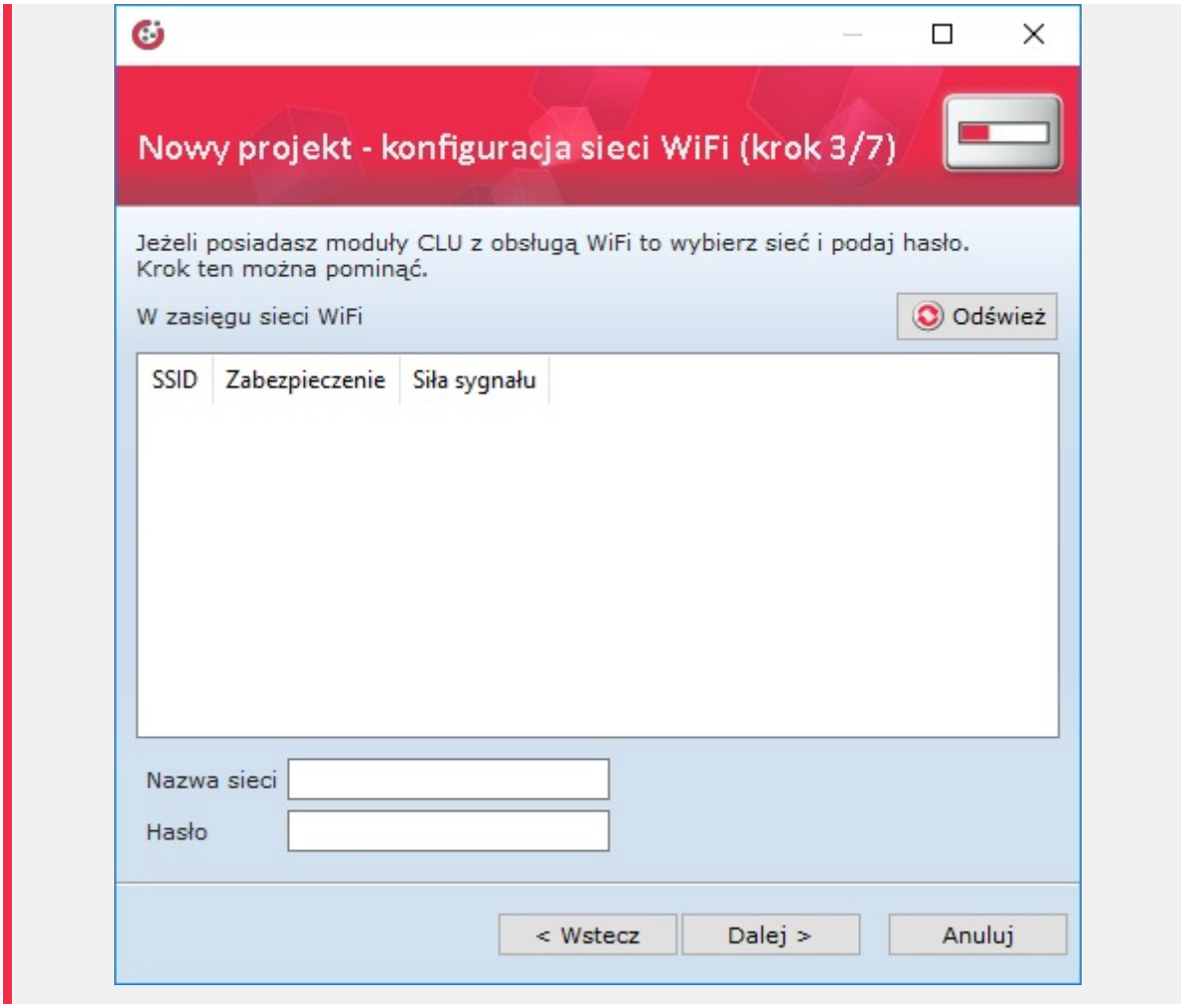
2. Program Object Manager wyświetli okno konfiguracji sieciowej. Ustawienia dla dostępnego interfejsu są wczytywane automatycznie. Pozostałe interfejsy sieciowe można wybrać z rozwijanej listy. Dodatkowo jest możliwe wskazanie zakresu, z jakiego system automatycznie przydzieli dostępne adresy IP znalezionym modułom.



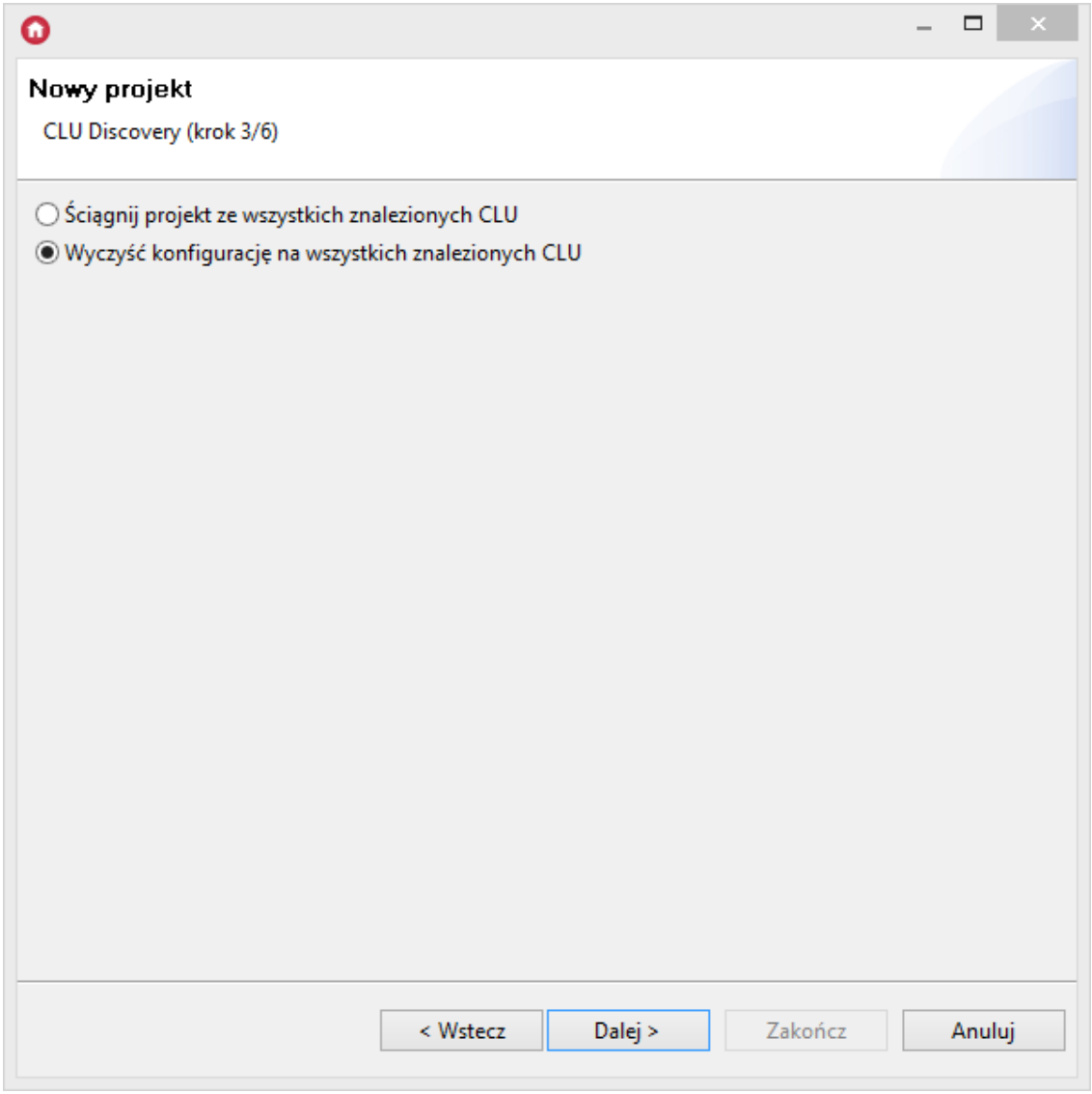
Uwaga! Dla Object Manager w wersji poniżej 1.2.1, ustawienia dla dostępnego interfejsu sieciowego nie są wczytywane automatycznie.



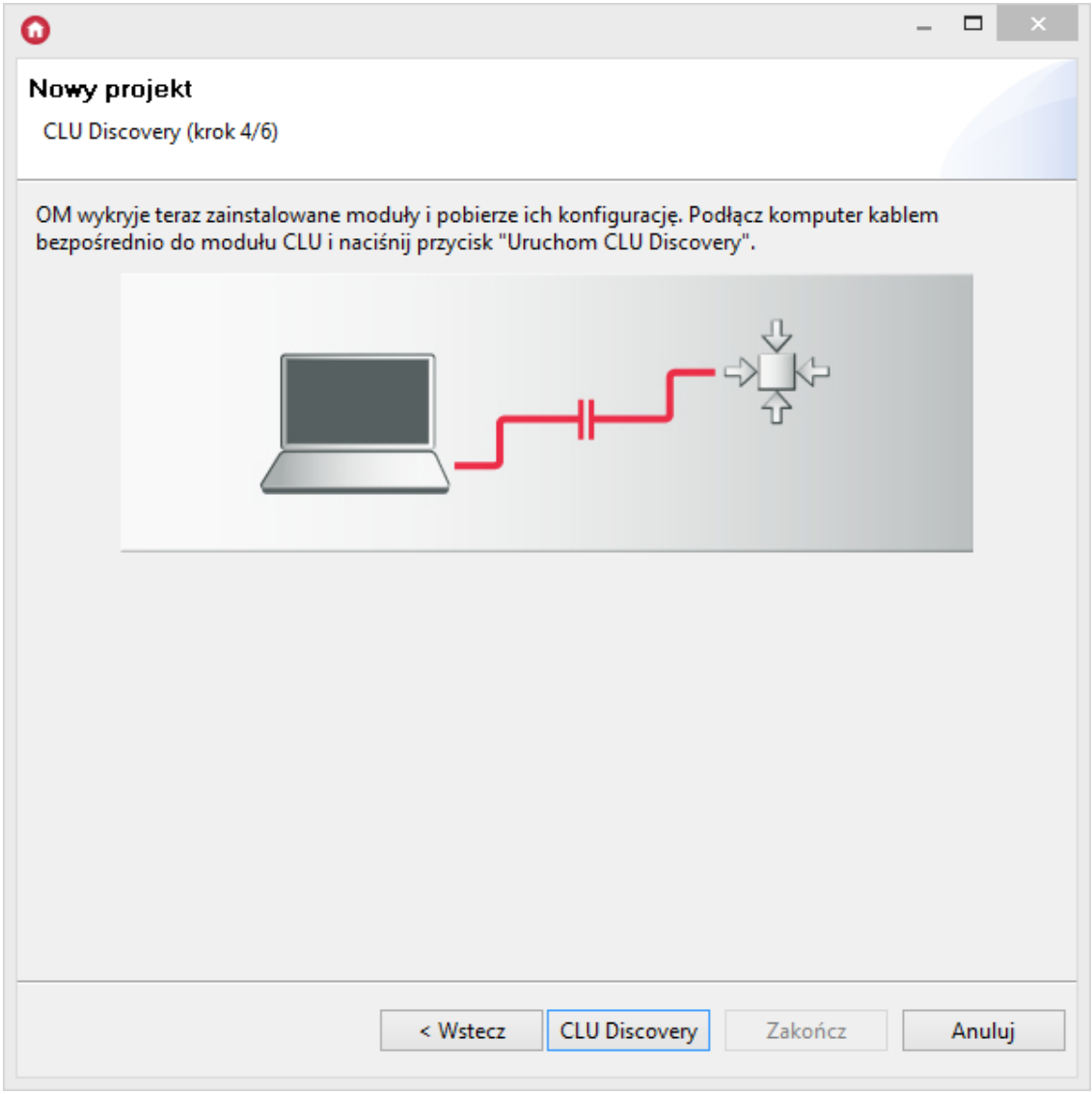
Uwaga! Dla starszej wersji Object Managera pojawia się krok dotyczący *konfiguracji sieci WiFi*, który należy pominąć.



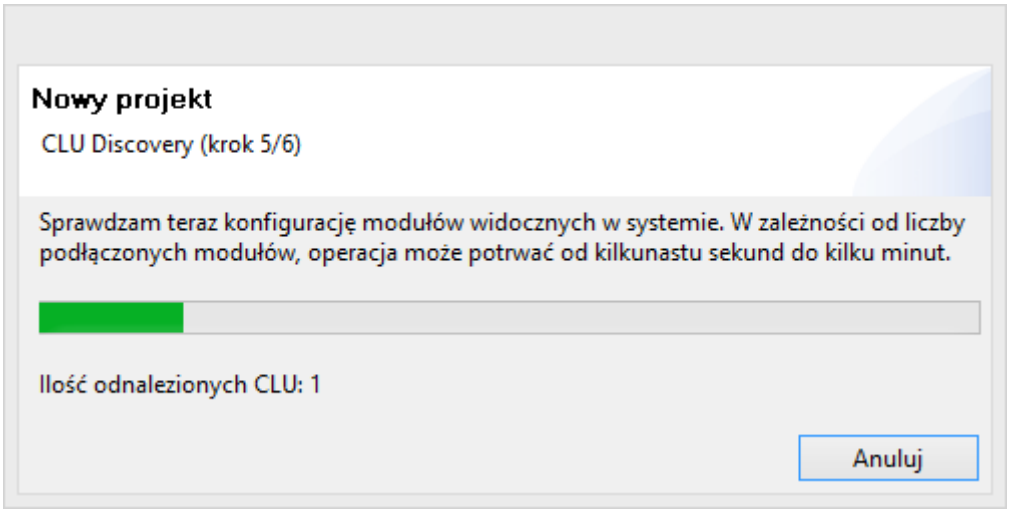
3. W kolejnym kroku można dokonać wyboru pomiędzy pobraniem istniejącej konfiguracji systemu do nowo tworzonego projektu, a całkowitym wyczyszczeniem konfiguracji i rozpoczęciem tworzenia projektu od podstaw. Pierwsza opcja jest przydatna w sytuacji konieczności odtworzenia konfiguracji po utracie pliku projektu.



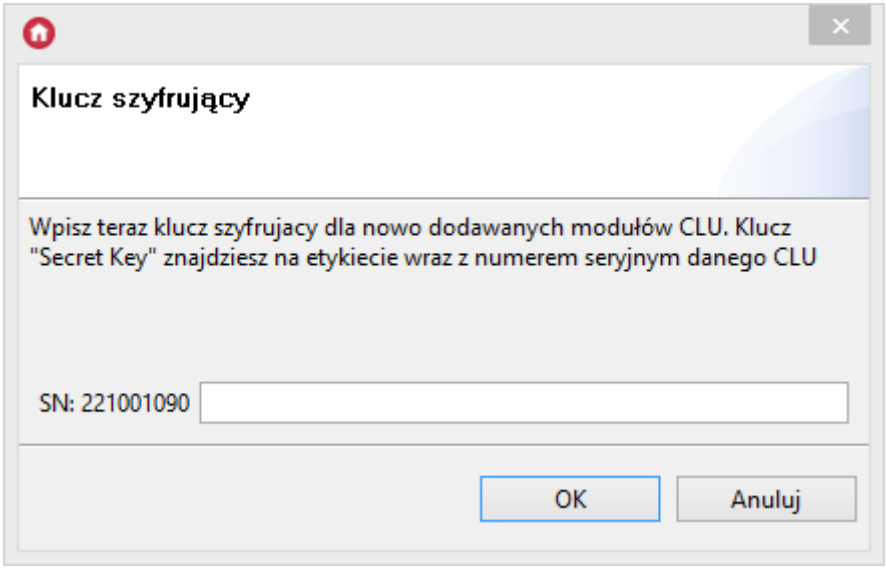
4. W czwartym kroku należy uruchomić procedurę wyszukiwania dostępnych modułów - CLU Discovery.



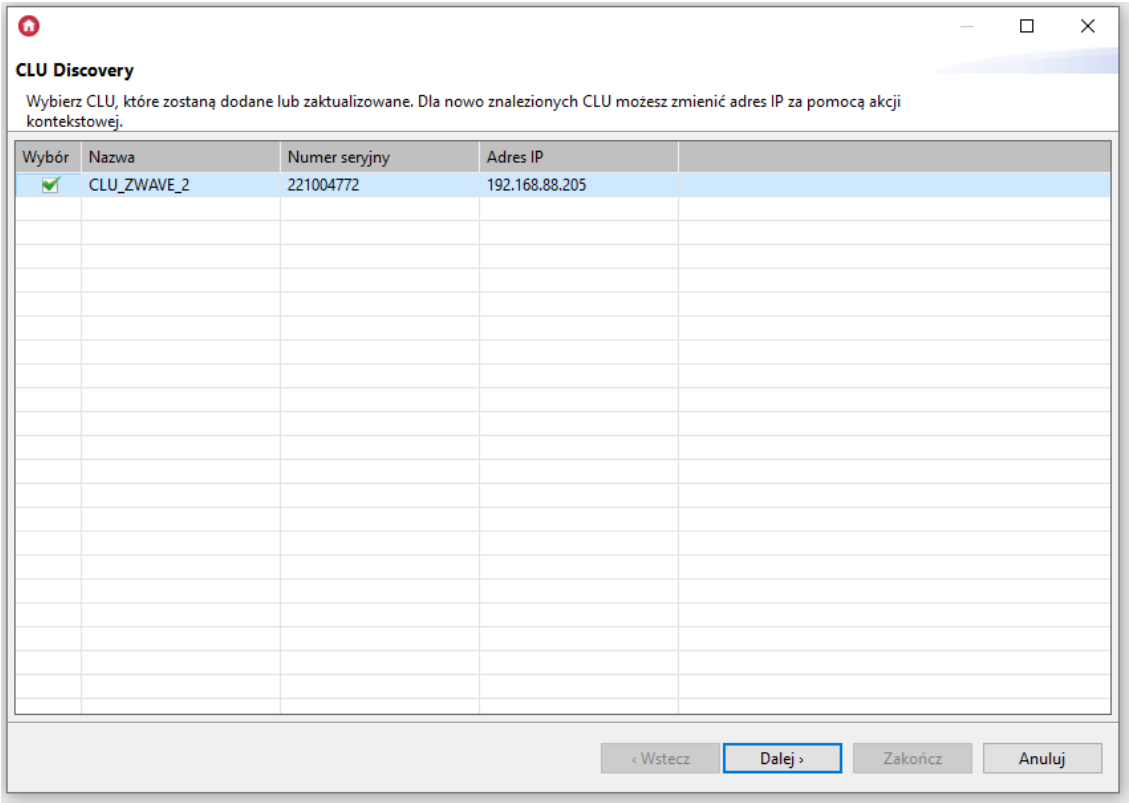
5. Object Manager rozpoczyna wyszukiwanie dostępnych modułów CLU.



Po odnalezieniu dostępnych modułów CLU należy podać ich klucze szyfrujące, które znajdziemy na obudowie danego modułu.

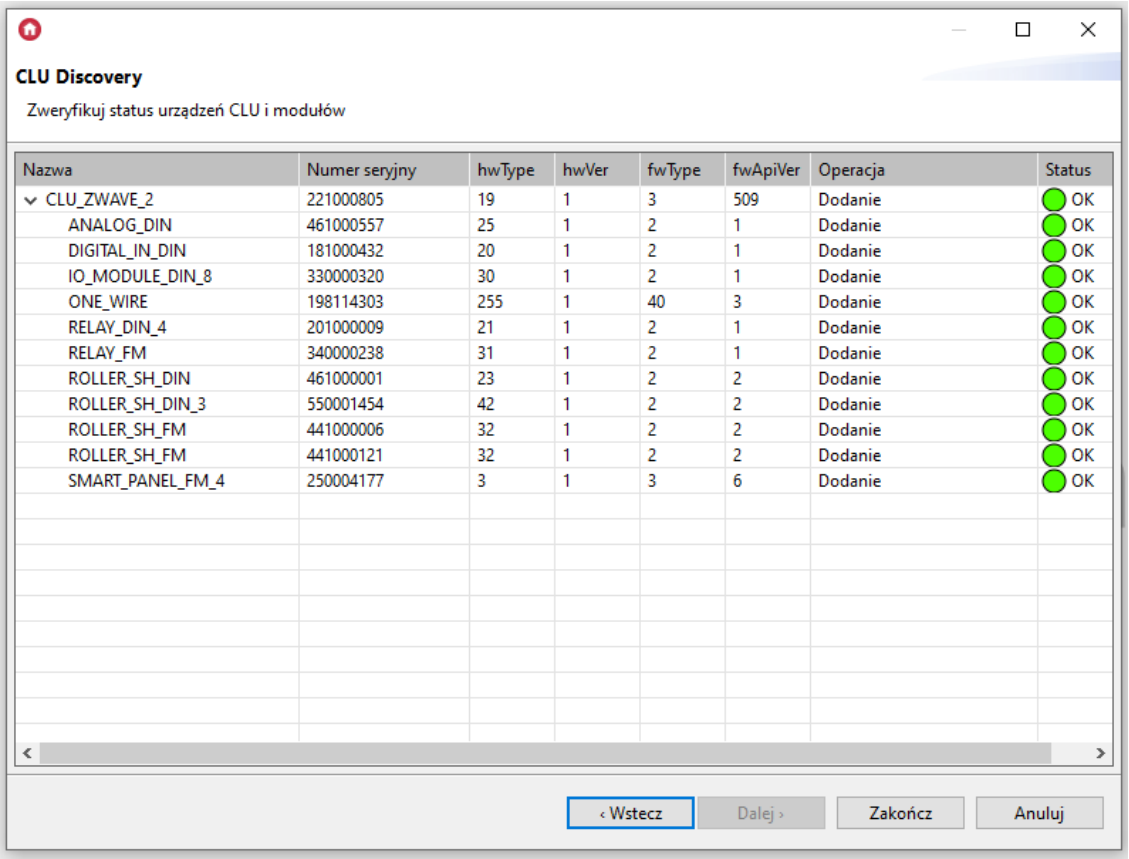


6. Po podaniu klucza dla odnalezionego CLU zostaje wyświetlone okno zawierające wszystkie CLU, które zostały znalezione podczas procedury CLU Discovery. Aby dany moduł CLU został dodany do projektu, musi zostać zaznaczone pole wyboru w kolumnie *Wybór*.



W tym oknie istnieje również możliwość zmiany adresu IP - opcja wywoływana jest z menu kontekstowego (PPM) dla danego (zaznaczonego w wierszu) CLU.

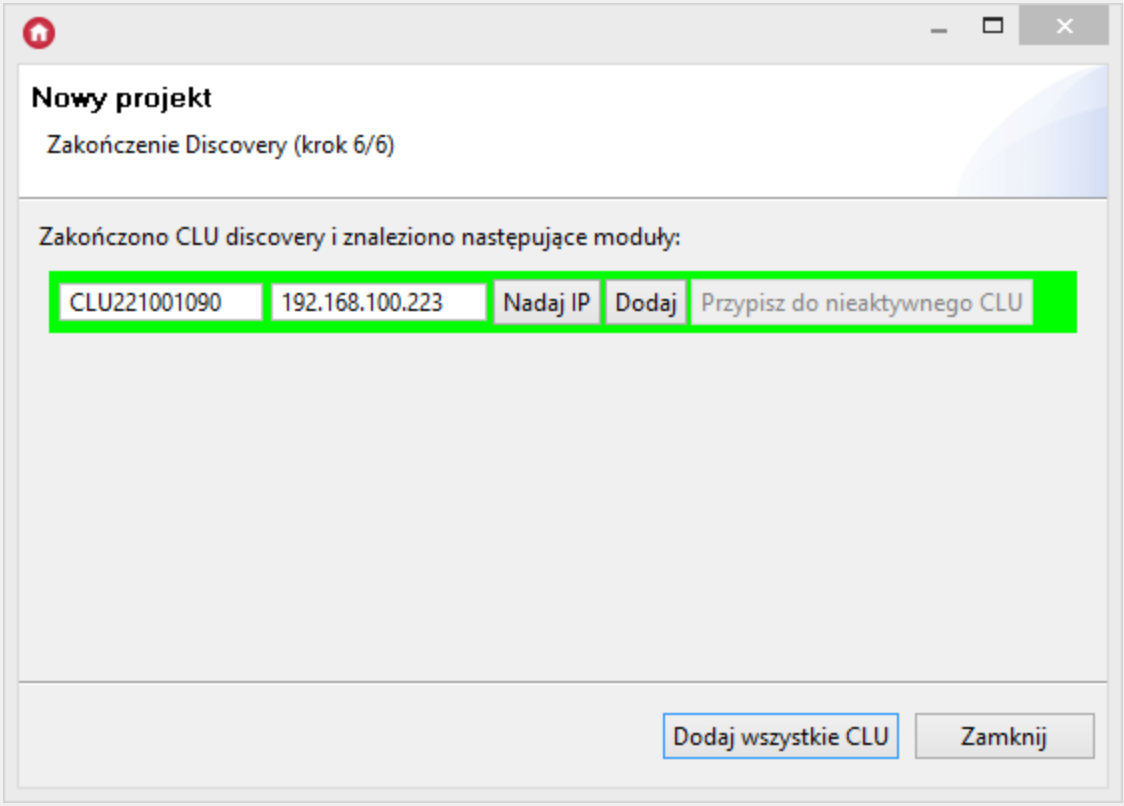
Przechodząc do następnego okna (przycisk *Dalej*) zostaje wyświetlona lista wszystkich wybranych (w poprzednim kroku) CLU wraz z modułami które są do niego podłączone (TF-Bus) / są sparowane (Z-Wave).



Opis wyświetlanych informacji w powyższym oknie umieszczono w rozdziale VI.4. Funkcja CLU Discovery.

Uwaga!

Dla Object Manager w wersji poniżej 1.6.0, nie są wyświetlane okna o których mowa w pkt.6 - wyświetlane zostaje okno z listą znalezionych modułów CLU. W tym oknie można dodać wszystkie lub wybrane moduły do tworzonego projektu. Istnieje również możliwość zmiany adresu IP, który został przydzielony automatycznie.



4. Funkcja CLU Discovery

Funkcja *CLU Discovery* w sposób całkowicie automatyczny wynajduje moduły CLU oraz podłączone do nich moduły IOM. Jest uruchamiana obligatoryjnie podczas otwierania nowego projektu, lecz może być również uruchomiona w każdej chwili ręcznie z menu akcji.



Z funkcji *CLU Discovery* należy korzystać, gdy:

- Podłączono do systemu nowy moduł CLU lub IOM;
- Dokonano wymiany modułu CLU lub IOM na inny;
- Przełączono moduł IOM z jednego CLU do innego;
- Istnieje potrzeba odzyskania całkowicie usuniętego obiektu IOM.

Po prawidłowym przeprowadzeniu CLU Discovery wszystkie zmiany zostaną wykryte i wprowadzone do projektu.

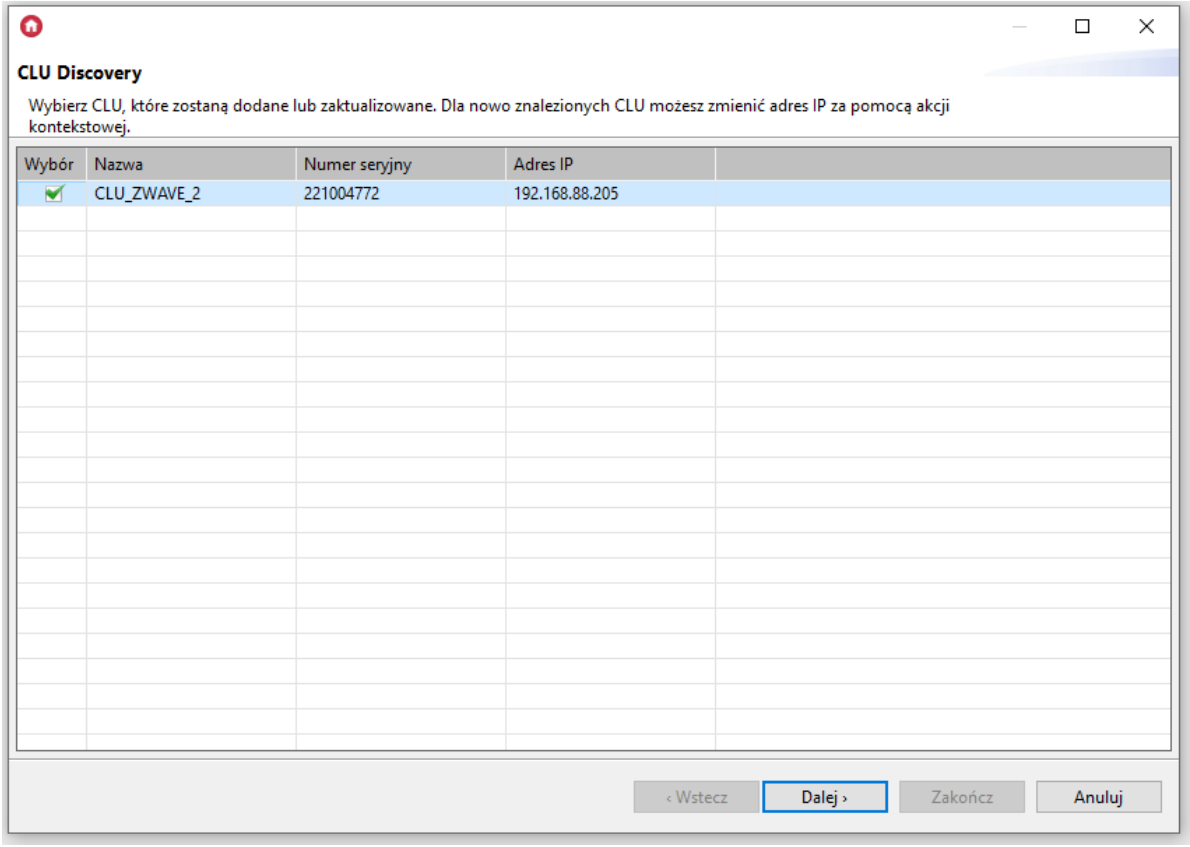
Przed rozpoczęciem CLU Discovery konieczne jest sprawdzenie czy:

- Wszystkie moduły są prawidłowo podłączone i zasilone;
- Moduły CLU są ze sobą połączone;
- Komputer, na którym uruchomiony jest OM, jest podłączony do tej samej sieci co CLU.

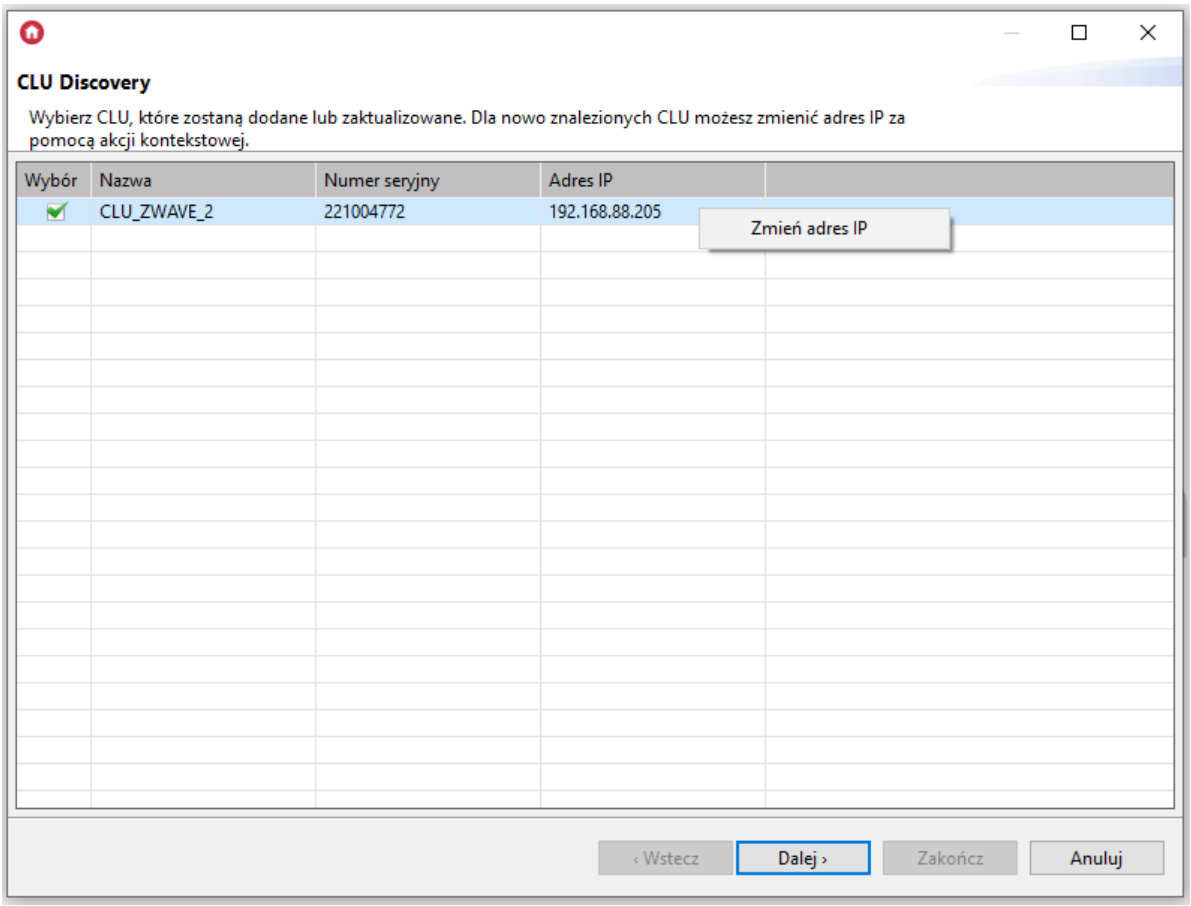
> **Uwaga!** Jeżeli w sieci znajduje się router, zalecamy, by podczas procesu CLU Discovery komputer był podłączony kablem sieciowym bezpośrednio z CLU. W większości przypadków CLU Discovery zakończy się sukcesem również w przypadku połączenia za pośrednictwem routera, jednak w przypadku specyficznej konfiguracji routera CLU Discovery może nie wykryć modułów CLU.

4.1. Dodanie modułów do projektu

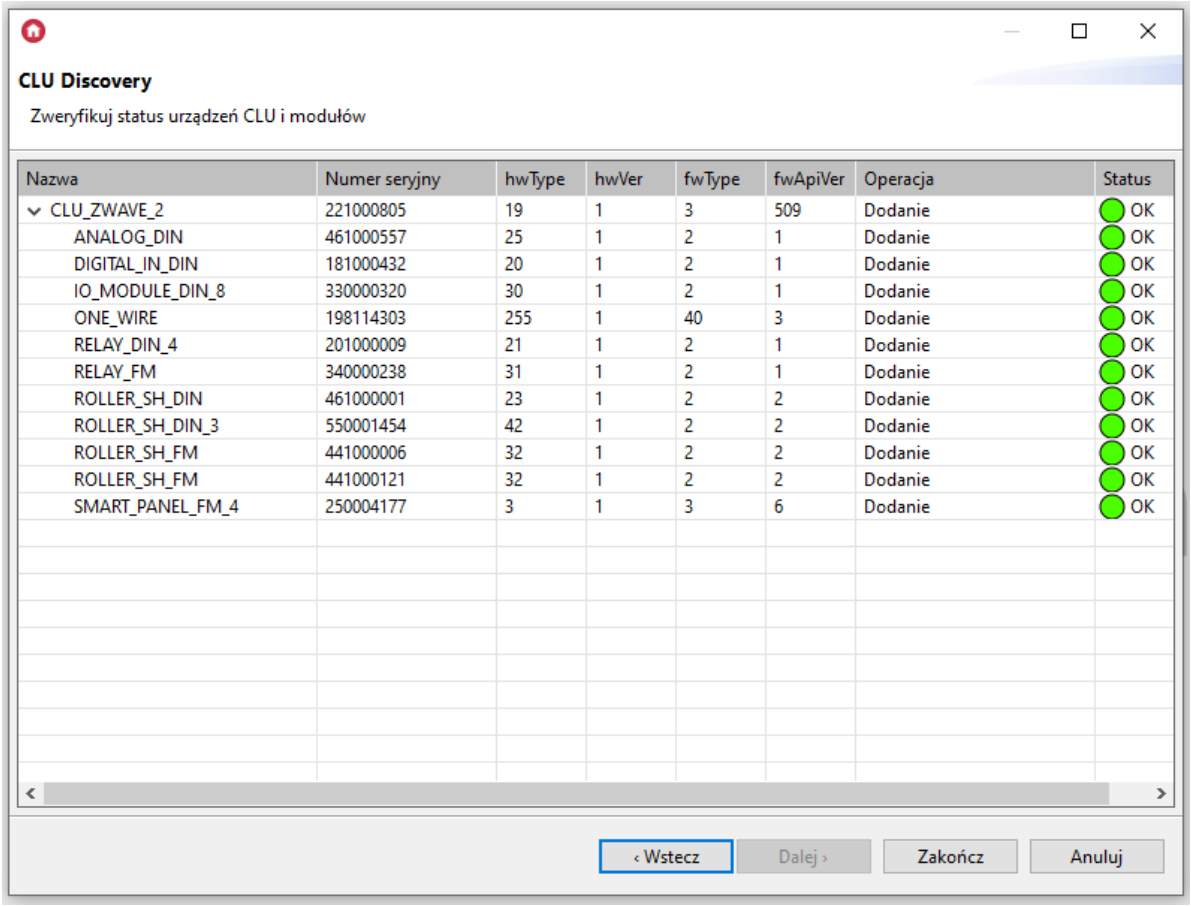
Wyświetlone okno zawierają wszystkie CLU, które zostały znalezione podczas procedury CLU Discovery. Aby dany moduł CLU został dodany do projektu, musi zostać zaznaczone pole wyboru w kolumnie *Wybór*.



W tym oknie istnieje również możliwość zmiany adresu IP - opcja wywoływana jest z menu kontekstowego (PPM) dla danego (zaznaczonego w wierszu) CLU. Opcja zmiany adresu IP dostępna jest tylko dla nowo znalezionych modułów CLU.



Przechodząc do następnego okna (przycisk *Dalej*) zostaje wyświetlona lista wszystkich wybranych (zaznaczonych w poprzednim kroku) CLU wraz z modułami które są do niego podłączone (TF-Bus) lub są sparowane (Z-Wave).



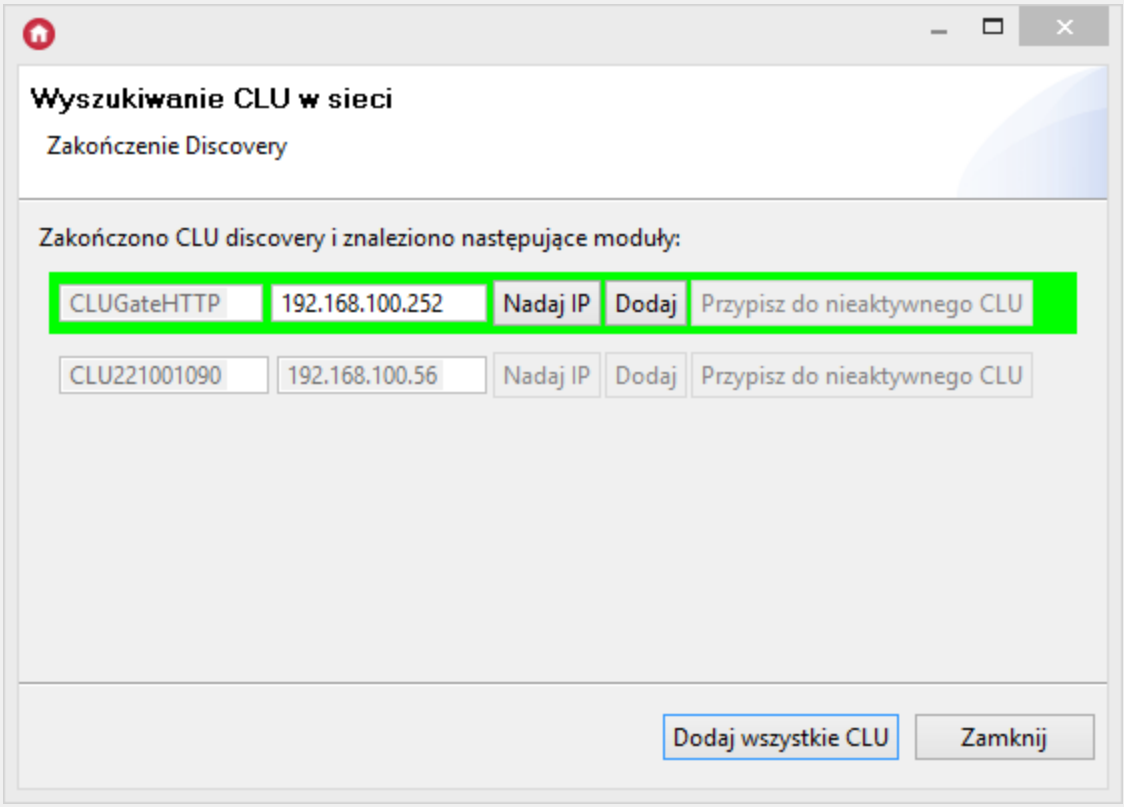
W oknie wyświetlane są następujące informacje:

- **Nazwa** - nazwa danego urządzenia (CLU / modułu);
- **Numer seryjny** - numer danego urządzenia nadany fabrycznie lub podczas procesu parowania (moduły Z-Wave);
- **hwType, hwVer, fwType, fwApiVer** - parametry konfiguracyjne danego urządzenia;
- **Operacja** - informacja dotycząca jakie działanie zostanie wykonane dla danego urządzenia (CLU / modułu):
    - **Dodanie** - nowy moduł zostaje dodany do projektu;
    - **Dodanie (Przypisanie)** - następuje przepisanie konfiguracji pomiędzy nieaktywnym a aktywnym modułem (dotyczy sytuacji, w której CLU Discovery uruchomiono na istniejącym projekcie);
    - **Aktualizacja** - obiekty danego urządzenia zostaną zaktualizowane - zgodnie z przeprowadzoną aktualizacją danego modułu (zmiana fwApiVer). W przypadku aktualizacji w kolumnie fwApiVer zostają wyświetlone dwie wartości: aktualna oraz sprzed procesu aktualizacji modułu (dotyczy sytuacji, w której CLU Discovery uruchomiono na istniejącym projekcie);
    - **Usunięcie** - w przypadku gdy dany moduł (TF-Bus / Z-Wave) nie zostanie odnaleziony podczas CLU Discovery, następuje wyszarzenie obiektów danego modułu w projekcie (dotyczy sytuacji, w której CLU Discovery uruchomiono na istniejącym projekcie);
    - **Brak** - brak zmian w konfiguracji danego modułu (dotyczy sytuacji, w której CLU Discovery uruchomiono na istniejącym projekcie).
- **Status** - informujący o możliwości przeprowadzenia akcji podanych w kolumnie *Operacja*:
    - **OK** - poprawne wykonanie danej operacji na module;
    - **Brak interfejsów XML** - dana operacja nie może zostać wykonana, ponieważ dla danego modułu brakuje interfejsu XML.

Jeżeli wszystkie moduły posiadają **Status: OK** - możliwe jest dodanie/wprowadzenie zmian dla danego CLU - przy pomocy przycisku *Zakończ*.

W sytuacji, gdy dane urządzenie posiada Status **Brak interfejsów XML**, nie jest możliwe dodanie/wprowadzenie zmian dla danego CLU, do którego podłączony jest dany moduł. W takim przypadku należy zaktualizować Bazę interfejsów, a następnie ponownie wykonać CLU Discovery.

Uwaga! Dla Object Manager w wersji poniżej 1.6.0, nie są wyświetlane wyżej wymienione okna (wybór CLU oraz statusu urządzeń) - wykryte moduły CLU zostają wyświetlone w postaci listy.



Kolor pozycji oznacza:

- **Zielony** – nowo wykryte CLU, które można dodać do projektu;
- **Czerwony** – CLU, które z różnych przyczyn nie może zostać dodane do projektu (wersja nieobsługiwana przez OM itp.);
- **Brak koloru** – CLU dodane wcześniej do projektu (dotyczy sytuacji, w której CLU Discovery uruchomiono na istniejącym projekcie).

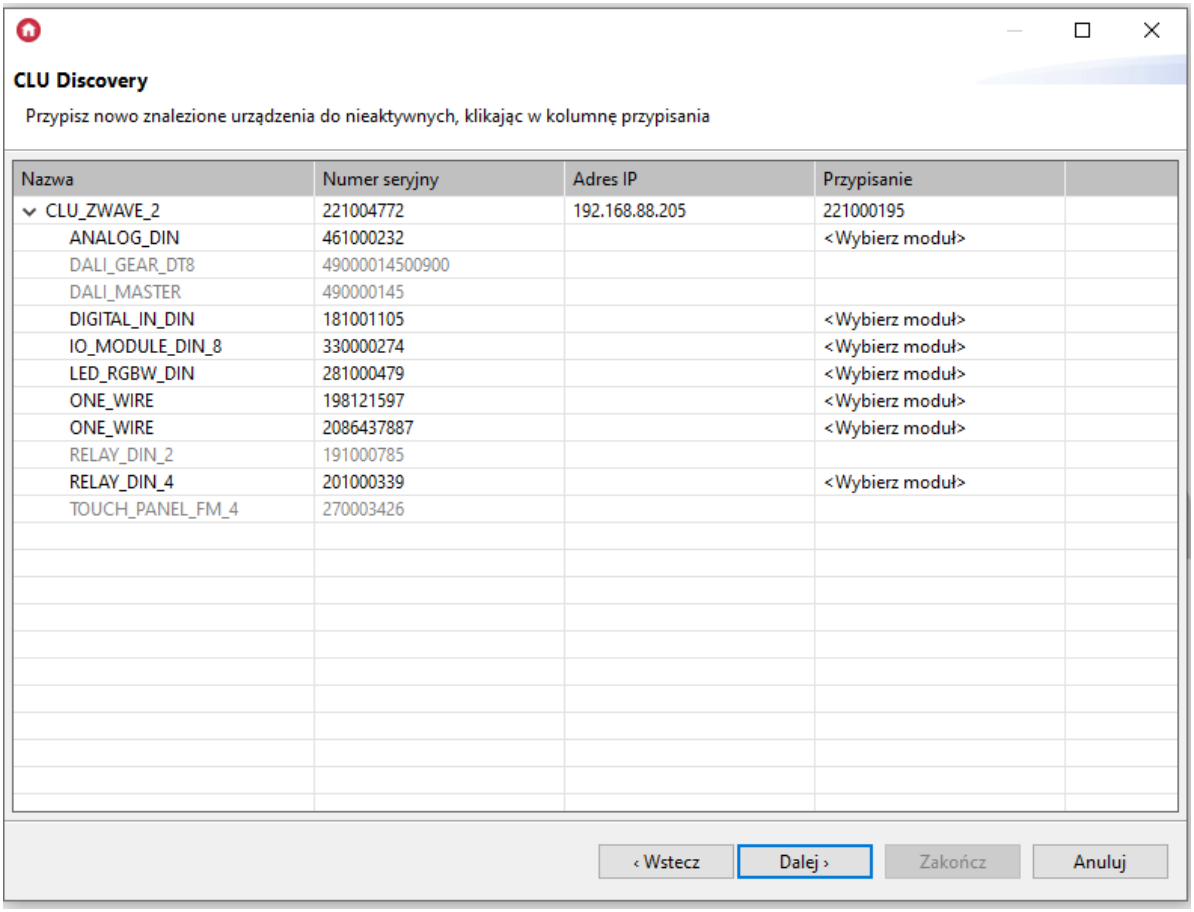
Moduły można dodać pojedynczo wciskając przycisk *Dodaj* lub za jednym razem wszystkie przy pomocy przycisku *Dodaj wszystkie CLU*.

Po wykonaniu powyższych czynności do projektu zostaną dodane wskazane CLU.

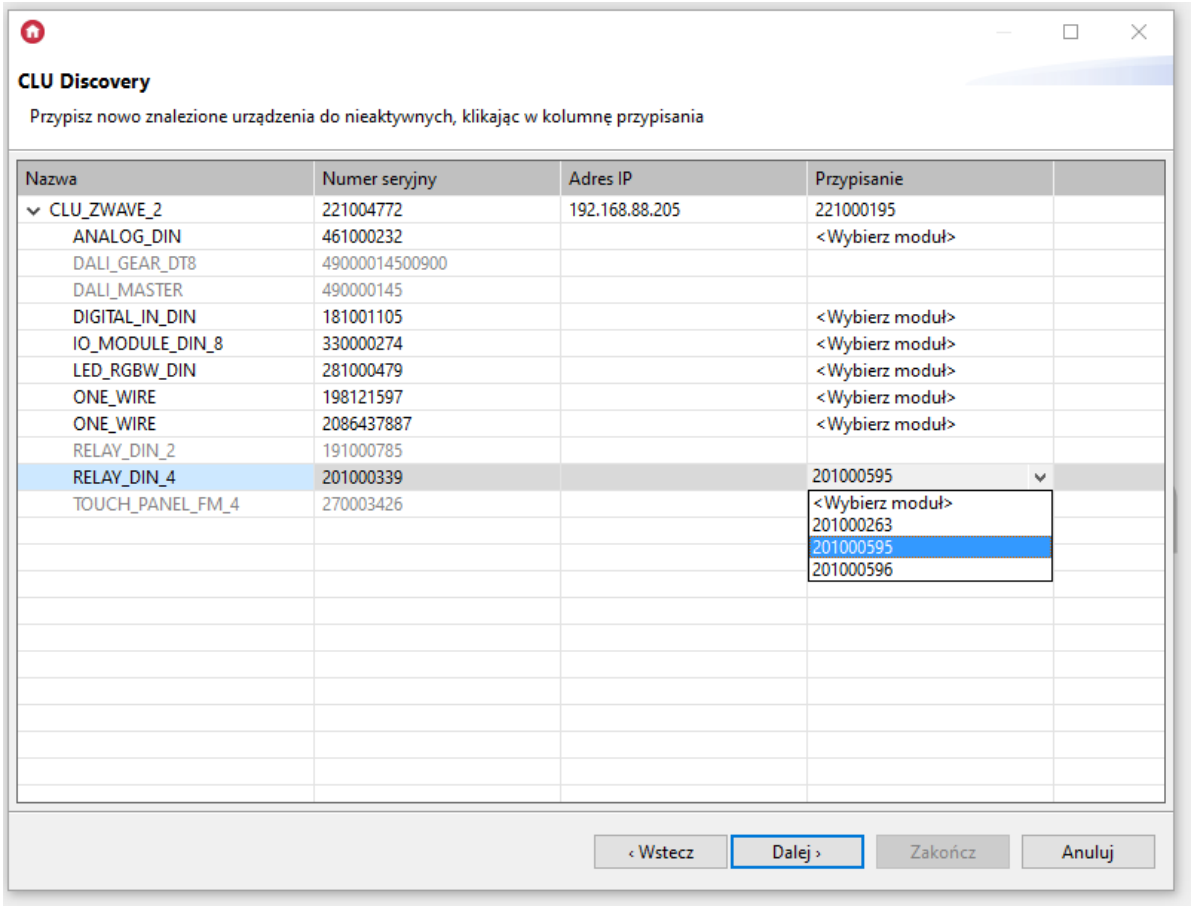
4.2. Podmiana / przypisywanie modułów podczas Discovery

Uwaga! Możliwość przypisywania modułów podczas CLU Discovery dostępna jest dla Object Managera w wersji 1.7.0 lub wyższej.

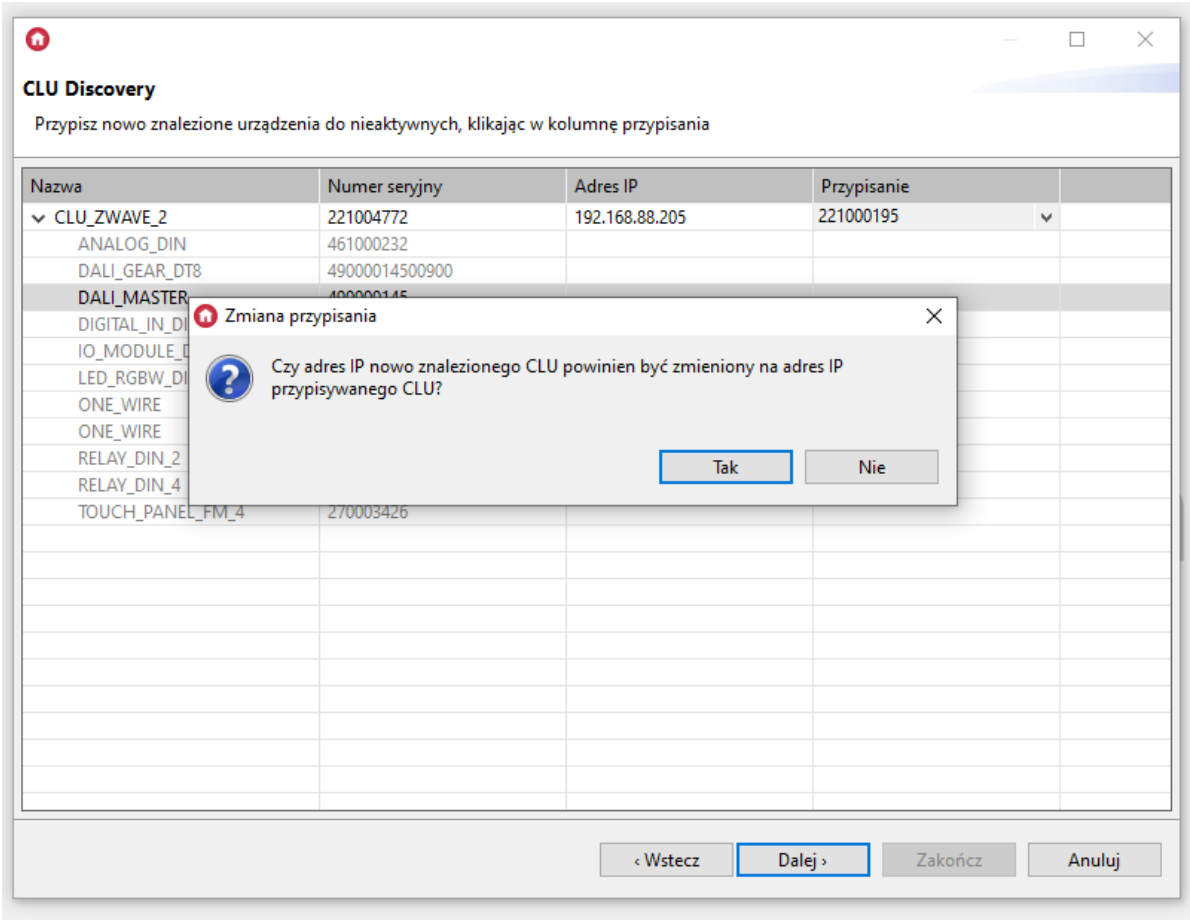
W przypadku fizycznej podmiany modułów, podczas procesu Discovery istnieje możliwość wykonania przypisania konfiguracji nieaktywnego (odłączonego) modułu do nowego dodanego do instalacji. Przypisanie odbywa się w ramach całego modułu (jego obiektów), bez konieczności wykonywania przypisywania pojedynczych obiektów danego modułu.



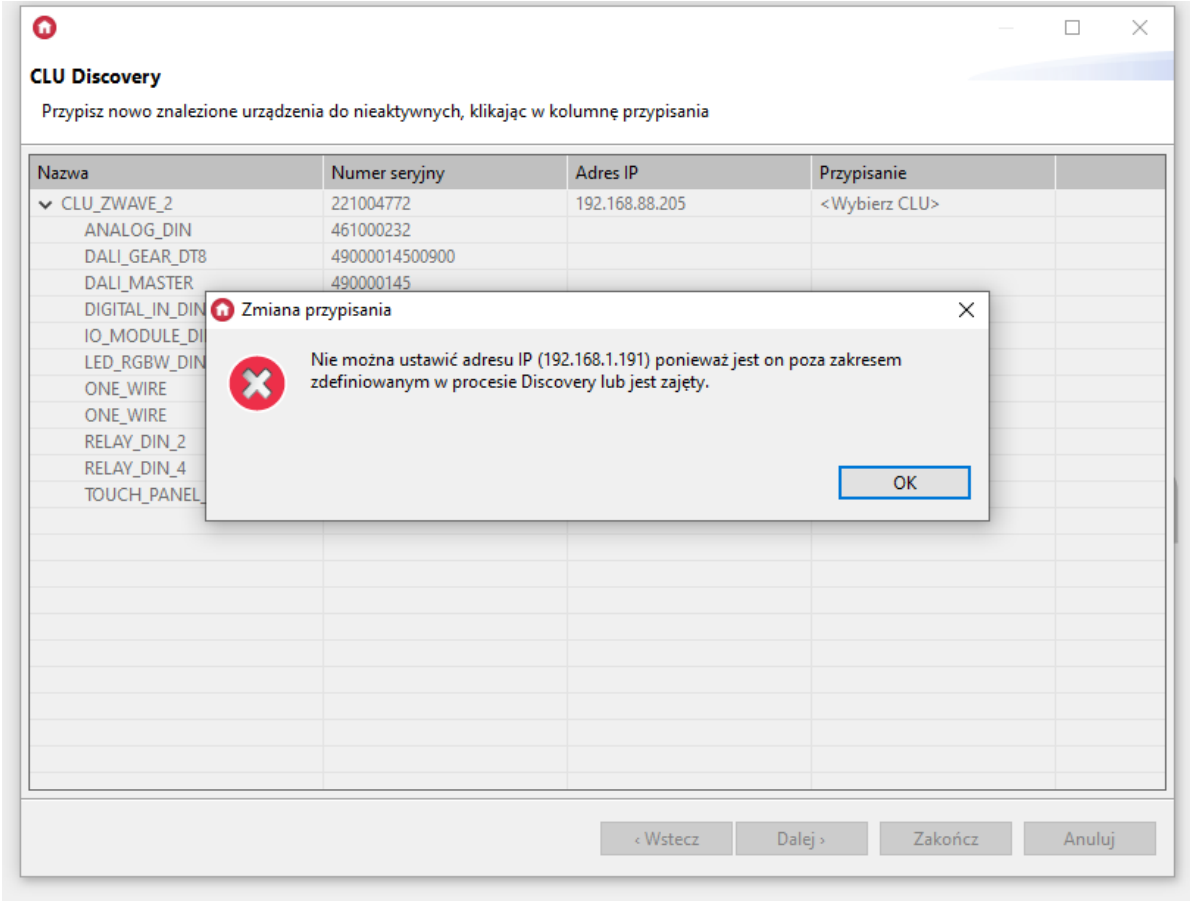
Okno pojawia się tylko w momencie kiedy istnieją możliwe przypisania pomiędzy modułami tego samego typu. W widoku prezentowane jest nowo znalezione CLU i/lub moduły oraz możliwe dla nich przypisania. Urządzenia prezentowane są w postaci hierarchicznej `CLU->Moduły`. W przypadku modułów, które nie mają dopasowania do przypisania, są one wyszarzane. Dla modułów mających dopasowanie, w kolumnie *Przypisanie* dla danego modułu wyświetlana jest lista nieaktywnych modułów których konfiguracja może zostać przepisana.



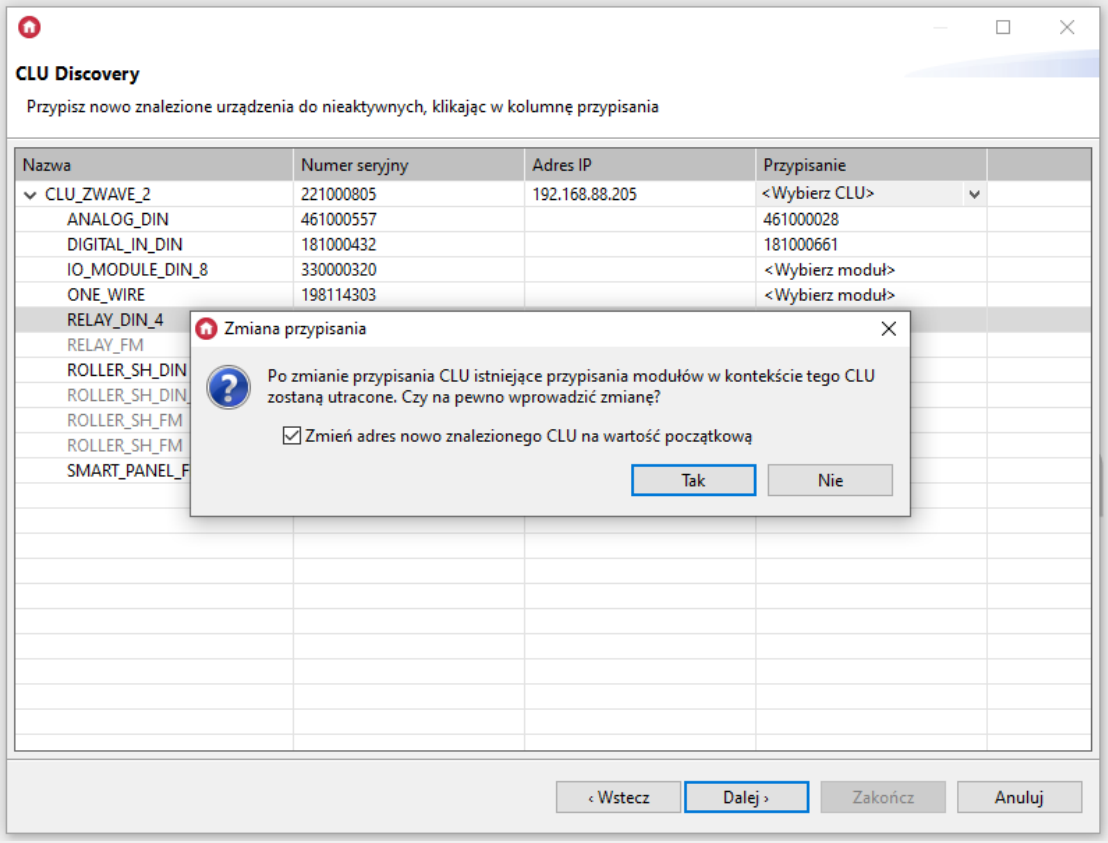
W przypadku przypisywania konfiguracji CLU, po wybraniu nieaktywnego modułu, wyświetlony zostaje komunikat o wykonaniu zmiany adresu IP na adres nieaktywnego CLU - zmiana zostanie wykonana jeżeli dany adres jest dostępny.



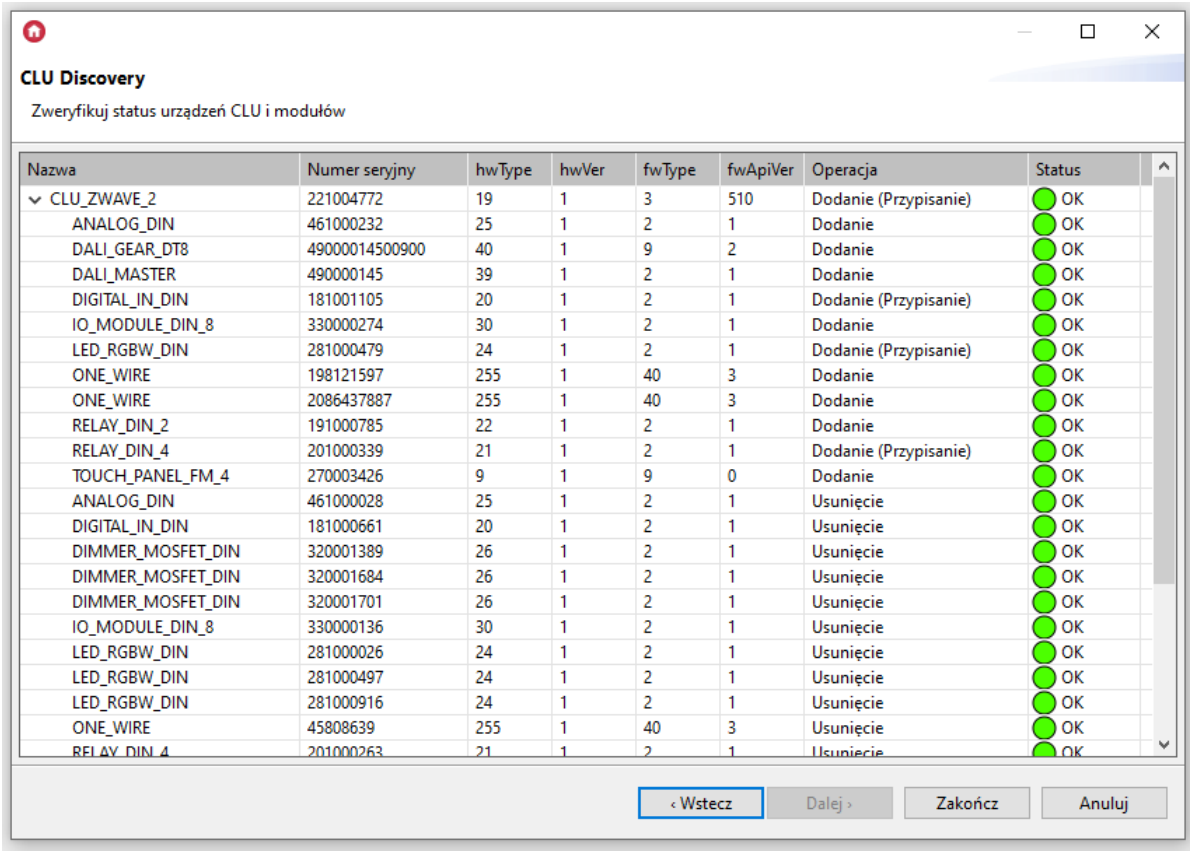
W przypadku zajętości / braku dostępu do adresu, zostaje wyświetlony adekwatny komunikat a dla CLU pozostaje nadany adres IP podczas procesu Discovery.



Uwaga! W przypadku usunięcia przypisania dla CLU, następuje usunięcie wykonanego przypisania pomiędzy modułami danego CLU, oraz możliwe jest przywrócenie poprzedniego (nadanego podczas Discovery) adresu IP (jeżeli został zmieniony podczas przypisywania) - użytkownik zostaje poinformowany następującym komunikatem:



Po dokonaniu przypisywania w kolejnym kroku (podsumowanie Discovery) przypisane modułu posiadają status *Dodanie (Przypisanie)*.



Kliknięcie przycisku *Zakończ* powoduje wysłanie konfiguracji do CLU.

5. Statusy CLU

5.1. Diody modułu

Na podstawie diod modułu CLU - użytkownik zostaje poinformowany, jaki jest aktualny status zarówno zasilania, konfiguracji jak i trybu w jakim aktualnie znajduje się urządzenie.

Status	Opis
	Brak zasilania
	Zielona dioda mruga co 500ms - system OK
	Błąd konfiguracji, system nieskonfigurowany lub brak komunikacji z modułem IOM
	Zielona dioda mruga co 200ms - CLU w trybie dodawania modułów Z-Wave
	Czerwona dioda mruga co 200ms - CLU w trybie usuwania modułów Z-Wave
	Dioda zielona świeci przez 1 sekundę, następnie obydwie diody mrugają trzykrotnie (co 200ms) - potwierdzenie dodania modułu Z-Wave
	Obydwie diody mrugają trzykrotnie (co 200ms), następnie czerwona gaśnie a zielona dioda mruga co 500ms - potwierdzenie usunięcia modułu Z-Wave
	Obydwie diody mrugają co 700ms - CLU w trybie logowania

5.2. Ikona modułu CLU w OM

Na podstawie wyglądu ikony modułu CLU - znajdującego się w menu obiektów otwartego projektu - użytkownik zostaje poinformowany, jaki jest aktualny status zarówno konfiguracji, jak i połączenia pomiędzy OM i CLU. Dla każdego CLU znajdującego się w projekcie można wyróżnić cztery tryby pracy: normalny, niepodłączony, błąd konfiguracji oraz tryb emergency.

Tryb normalny

CLU znajdujące się w normalnym trybie pracy nie zawiera błędów konfiguracyjnych, a połączenie pomiędzy OM i CLU jest aktywne. Nazwa tego modułu jest wyświetlana w kolorze czarnym, natomiast ikona obrazująca ten stan wygląda następująco:



Jeżeli nazwa danego CLU poprzedzona jest symbolem `*` oznacza to, iż w projekcie dokonano zmiany w konfiguracji, która nie została jeszcze wysłana do tego CLU.

Niepodłączony

Jeśli pomiędzy modułem CLU i OM nie ma połączenia (brak fizycznego podłączenia lub błąd w konfiguracji LAN), to nazwa CLU wyświetlona będzie w kolorze czerwonym, a ikona tego modułu będzie wyglądać następująco:



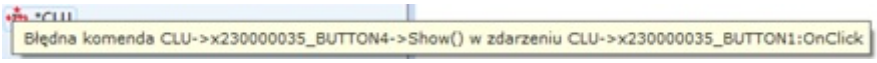
Jeśli CLU jest w stanie niepodłączonym, użytkownik ma możliwość dokonywania oraz zapisywania zmian w projekcie, natomiast nowa konfiguracja nie zostanie wysłana do CLU - jest to możliwe tylko w trybie normalnym.

Błąd konfiguracji

Jeżeli podczas pracy nad projektem zostaną wprowadzone zmiany zawierające błędy konfiguracyjne (np. zostanie utworzone powiązanie z nieistniejącym obiektem lub zostaną wprowadzone nieobsługiwane komendy) to CLU, w którym został wykryty błąd, zostanie przełączone w tryb pracy `Błąd konfiguracji`. Nazwa tego CLU wyświetlana będzie w kolorze czarnym, ale przy ikonie zostanie wyświetlony symbol błędu:



Po najechaniu myszką na to CLU rozwinie się pole z listą wykrytych błędów.



> **Uwaga!** OM nie pozwala na wysłanie do CLU konfiguracji zawierającej błędy!

Tryb emergency

Jeżeli do CLU zostanie wysłana konfiguracja zawierająca błędy w składni (np. po wysłaniu skryptu w edycji tekstowej) lub w wyniku działania skryptu dojdzie do zawieszenia interpretera LUA, to wtedy CLU zmieni tryb pracy na TRYB EMERGENCY. Nazwa CLU zmieni kolor na pomarańczowy, a przy ikonie pojawi się symbol awarii:



Jeżeli CLU zostało wprowadzone w tryb emergency, należy sprawdzić poprawność ostatnio wprowadzonych zmian i wysłać ponownie konfigurację do CLU.

> **Uwaga!** Moduły CLU wyjęte z pudełka (w stanie fabrycznym) znajdują się w trybie *Emergency*!

6. Obsługa modułów Z-Wave

Bezprzewodowe moduły IOM komunikują się z pozostałymi elementami systemu z wykorzystaniem protokołu Z-Wave. Zarówno z poziomu OM jak i z punktu widzenia sterowania są rozpoznawane i działają dokładnie w taki sam sposób, jak pozostałe moduły w systemie GRENTON.

Korzystanie z modułów Z-Wave w systemie jest możliwe, gdy znajduje się w nim co najmniej jeden moduł CLU wyposażony w kontroler Z-Wave.

> **Uwaga!** Dodanie modułu Z-Wave do systemu powinno odbywać się po umieszczeniu go **w miejscu docelowym instalacji** - wynika to z wymagań dotyczących tworzenia sieci *mesh*, zasięgu działania urządzenia oraz występujących zakłóceń sieci Z-Wave.

6.1. Dodawanie modułów Z-Wave

By moduły IOM Z-Wave były obecne w systemie konieczne jest ich dodanie do CLU. Można to zrobić na dwa sposoby:

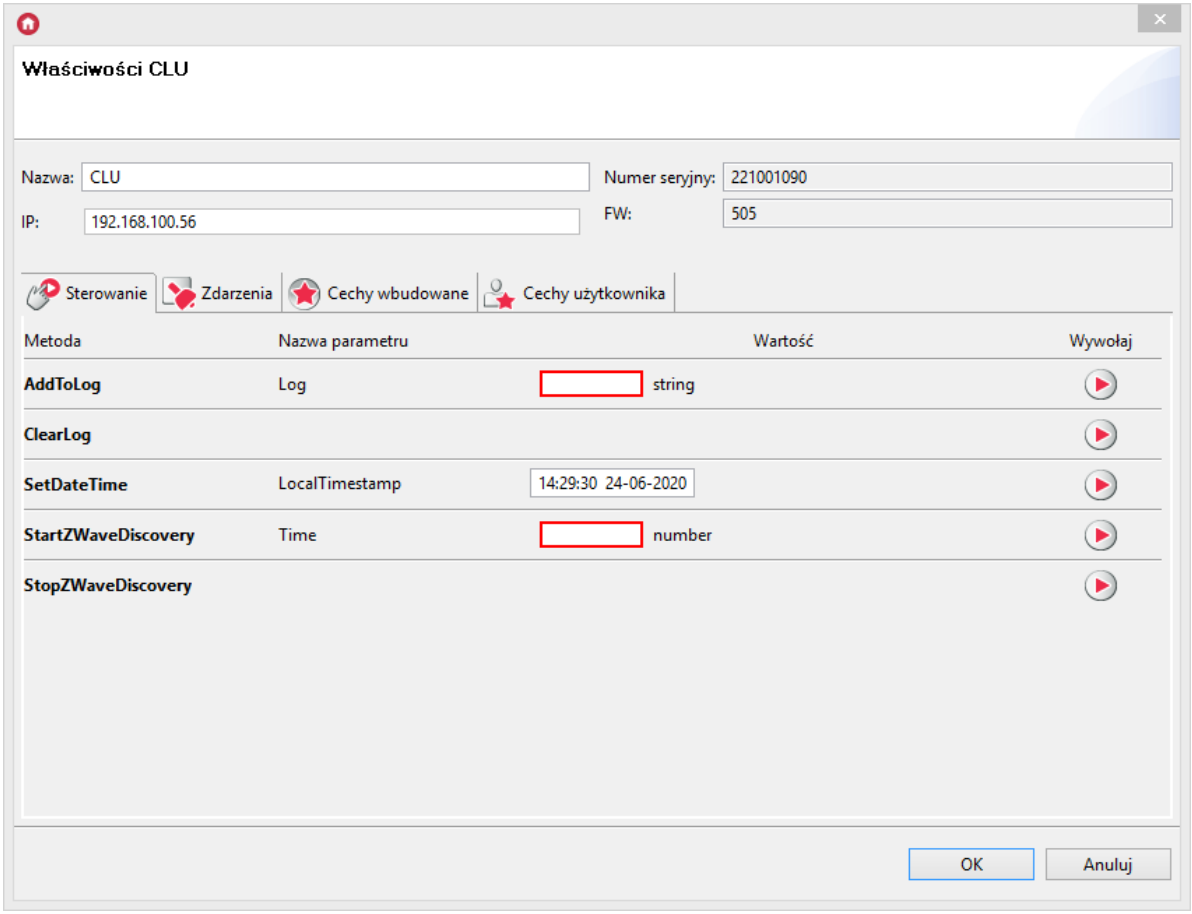
1. **Poprzez przycisk *LINK* znajdujący się na module CLU** W tym celu konieczne jest wciśnięcie przycisku `Link`, znajdującego się na module CLU z kontrolerem Z-Wave.

Po jego naciśnięciu CLU przechodzi do trybu dodawania modułów – dioda ON mruga cały czas w odstępach 200ms.

Później należy nacisnąć przycisk znajdujący się na dodawanym module Z-Wave (według instrukcji modułu). Prawidłowe dodanie modułu zostanie zasygnalizowane przez zapalenie na 1 sekundę diody ON, a następnie przez trzykrotne mrugnięcie diod ON oraz ERR w odstępach 200ms. Po zakończeniu dodawania modułu Z-Wave dioda ON zacznie migać z częstotliwością 500ms. Po zakończeniu dodawania modułów Z-Wave należy wykonać *CLU Discovery* - nowe moduły Z-Wave zostaną dodane do projektu.

2. **Za pomocą oprogramowania Object Manager** Taki sposób dodawania pozwala na zdefiniowanie czasu, przez jaki CLU będzie oczekiwało na „przedstawienie się" modułów bezprzewodowych. Rozwiązanie to jest bardzo przydatne, gdy dodawane moduły znajdują się w większej odległości od CLU i potrzeba więcej czasu na naciśnięcie znajdującego się na nich przycisku.

W celu dodania modułów bezprzewodowych przy pomocy OM należy otworzyć konfigurator obiektu dla modułu CLU Z-Wave, do którego będą dodawane moduły bezprzewodowe (dwukrotne kliknięcie w ikonę CLU na liście obiektów). Następnie w zakładce `Sterowanie` konieczne jest podanie czasu (jako parametr) dla metody `StartZWaveDiscovery` i wywołanie tej metody.



Podany czas, będzie czasem jaki CLU oczekuje na zgłoszenie się nowych modułów Z-Wave. Po jego upłynięciu wyszukiwanie zostaje zakończone, nawet jeśli żadne moduły nie zostały znalezione. Wpisanie cyfry *0* spowoduje, że wyszukiwanie zostanie zakończone automatycznie po znalezieniu pierwszego nowego modułu.

Po wywołaniu metody `StartZWaveDiscovery` należy nacisnąć przycisk znajdujący się na dodawanym module Z-Wave (zgodnie z instrukcją modułu). Poprawne dodanie modułu zostanie zasygnalizowane przez zapalenie na 1 sekundę diody ON, a następnie przez trzykrotne mrugnięcie diod ON oraz ERR w odstępach 200ms. Po poprawnym dodaniu modułów Z-Wave dioda ON zacznie migać z częstotliwością 500ms. Po zakończeniu dodawania modułów Z-Wave należy wykonać proces *CLU Discovery* - nowe moduły Z-Wave zostaną dodane do projektu.

Uwaga! Wywołanie metody `StopZWaveDiscovery` przerywa wyszukiwanie modułów Z-Wave.

Uwaga! Nie należy dodawać do systemu modułów, które już wcześniej zostały do niego podłączone. W przypadku braku pewności czy dany moduł został wcześniej dodany, w pierwszej kolejności należy wykonać procedurę usunięcia dla tego modułu.

Sytuacja ma się podobnie, gdy moduł Z-Wave był podłączony i nie został usunięty z innego kontrolera - najpierw należy wykonać procedurę usunięcia takiego modułu.

6.2. Usuwanie modułów Z-Wave

By moduł bezprzewodowy przestał się pojawiać w konfiguracji systemu musi on zostać usunięty z CLU Z-Wave.

W tym celu konieczne jest wciśnięcie przycisku `Unlink`, znajdującego się na module CLU z kontrolerem.

Po jego naciśnięciu CLU przechodzi do trybu usuwania modułów – dioda ERR mruga cały czas w odstępach 200ms.

Następnie należy nacisnąć przycisk znajdujący się na module bezprzewodowym (zgodnie z instrukcją modułu), który ma zostać usunięty. Prawidłowe usunięcie modułu zostanie zasygnalizowane przez trzykrotne mrugnięcie diod ON oraz ERR w odstępach 200ms. Po zakończeniu usuwania modułu Z-Wave dioda ERR zgaśnie, a ON zacznie migać z częstotliwością 500ms. Ostatnim krokiem będzie przeprowadzenie *CLU Discovery* - usunięte moduły zostaną wyszarzone.

6.3. Brak komunikacji z modułem Z-Wave - mechanizm zliczania niepowodzeń komunikacji i blokowania komunikacji urządzenia w sieci Z-Wave

> **Uwaga!** Przedstawiony mechanizm dostępny jest dla CLU od wersji **04.07.41(183201)**

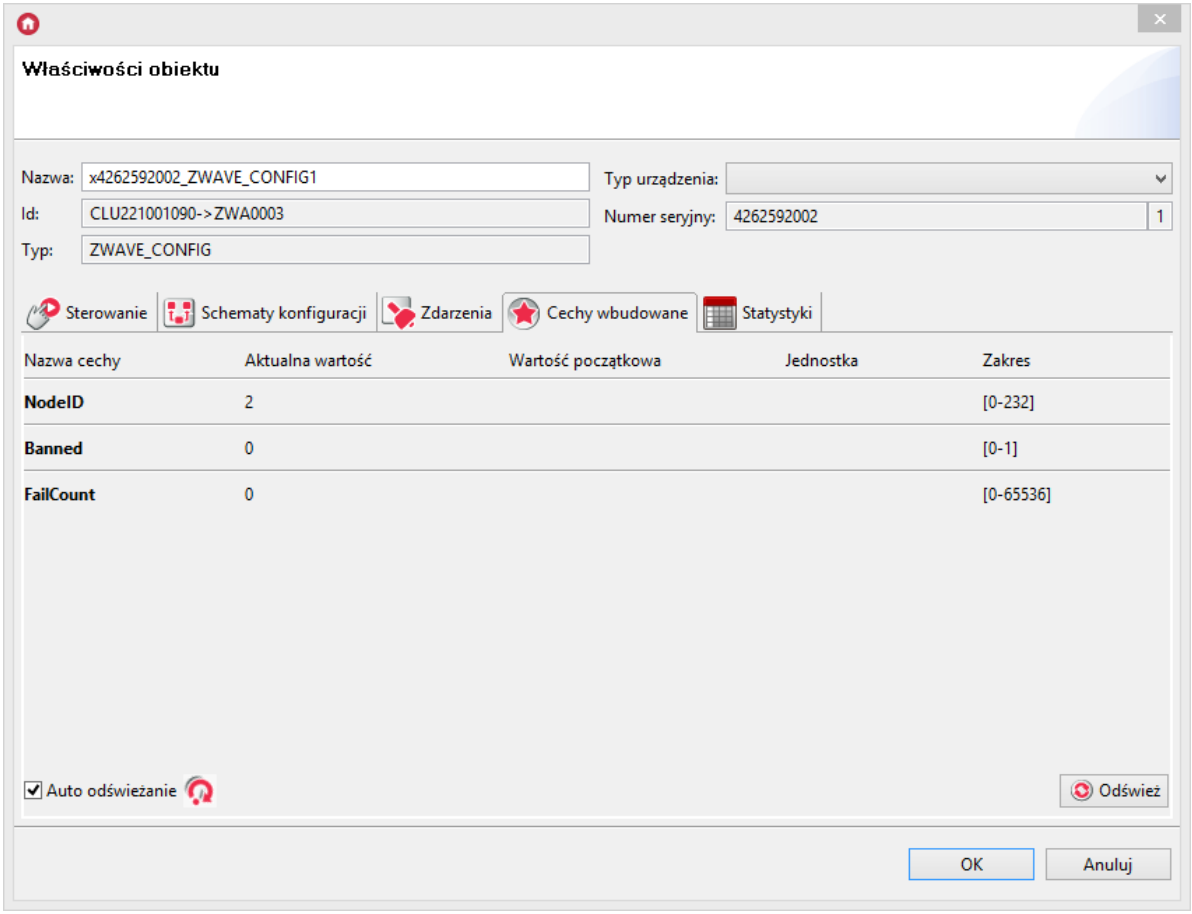
Niepowodzenia w komunikacji z urządzeniem Z-Wave mogą nastąpić w sytuacji, gdy:

- moduł Z-Wave jest uszkodzony,
- brak zasilania (230V) na module / wyczerpania baterii zasilającej moduł,
- urządzenie pracuje na pograniczu zasięgu z kontrolerem / nie jest w zasięgu kontrolera,
- kontroler (CLU) po wysłaniu rozkazu nie otrzyma potwierdzenia od urządzenia (ACK).

Informacje o statusie urządzenia w sieci Z-Wave można odczytać z pozycji Object Managera przy pomocy obiektu ZWAVE_CONFIG danego modułu Z-Wave.

> **Uwaga!** Obiekty ZWAVE_CONFIG nie są dostępne dla wszystkich modułów Z-Wave - posiadają je modułu Grenton Z-Wave oraz wybrane moduły, które są wspierane przez system Grenton.

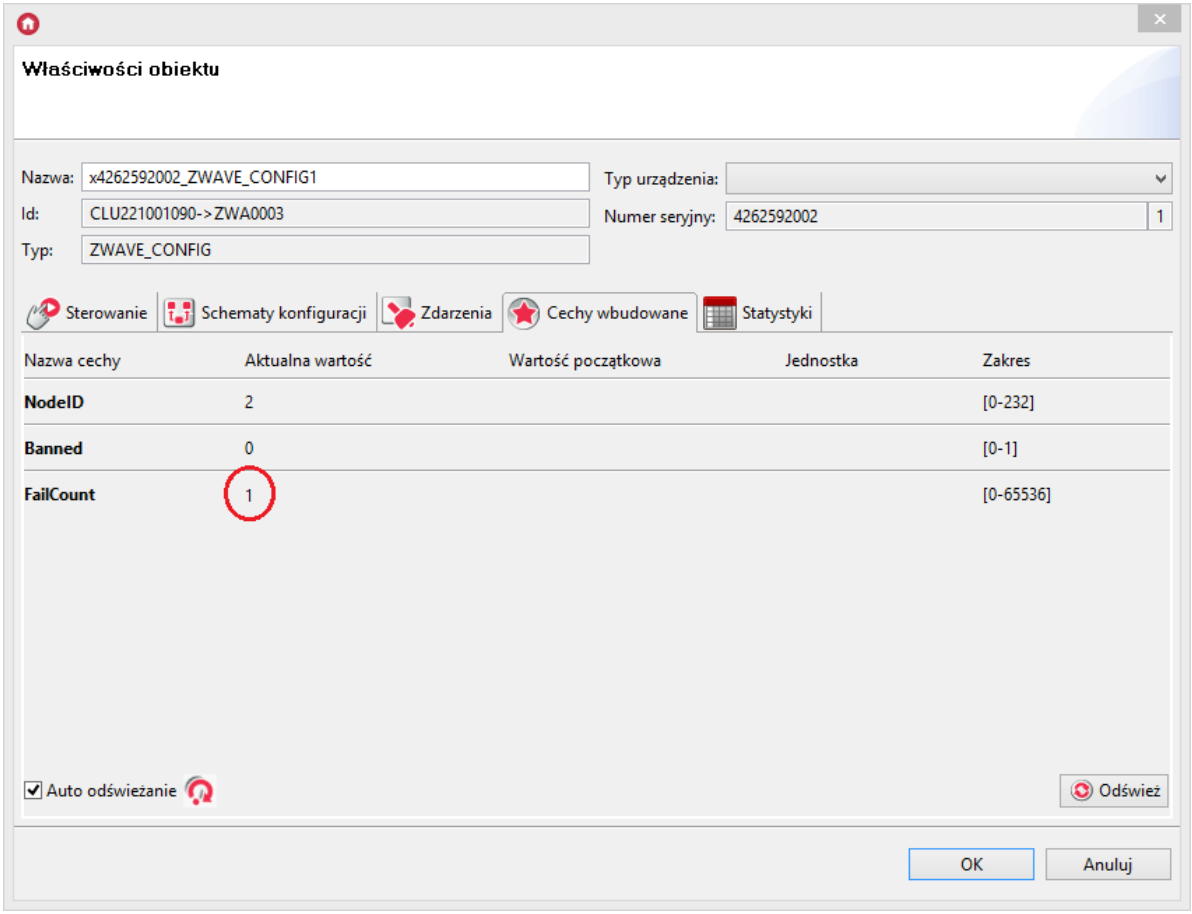
Dla danego obiektu dostępne są następujące cechy:



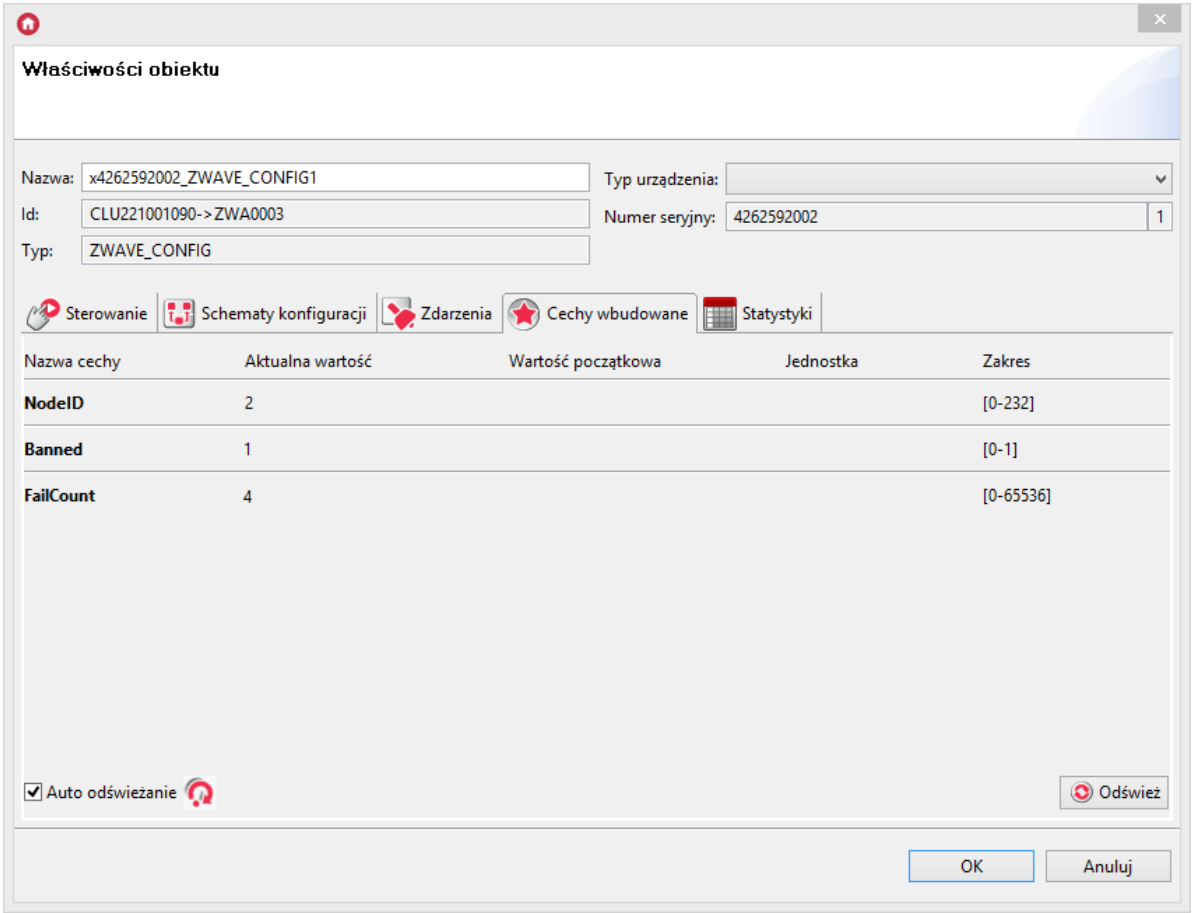
- `NodeID` - Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
- `Banned` - Informacja o zablokowaniu komunikacji Z-Wave z modułem
- `FailCount` - Liczba nieudanych prób komunikacji z modułem Z-Wave

Mechanizm zliczania niepowodzeń w komunikacji:

- W przypadku wystąpienia niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy `FailCount` obiektu ZWAVE_CONFIG danego urządzenia Z-Wave.

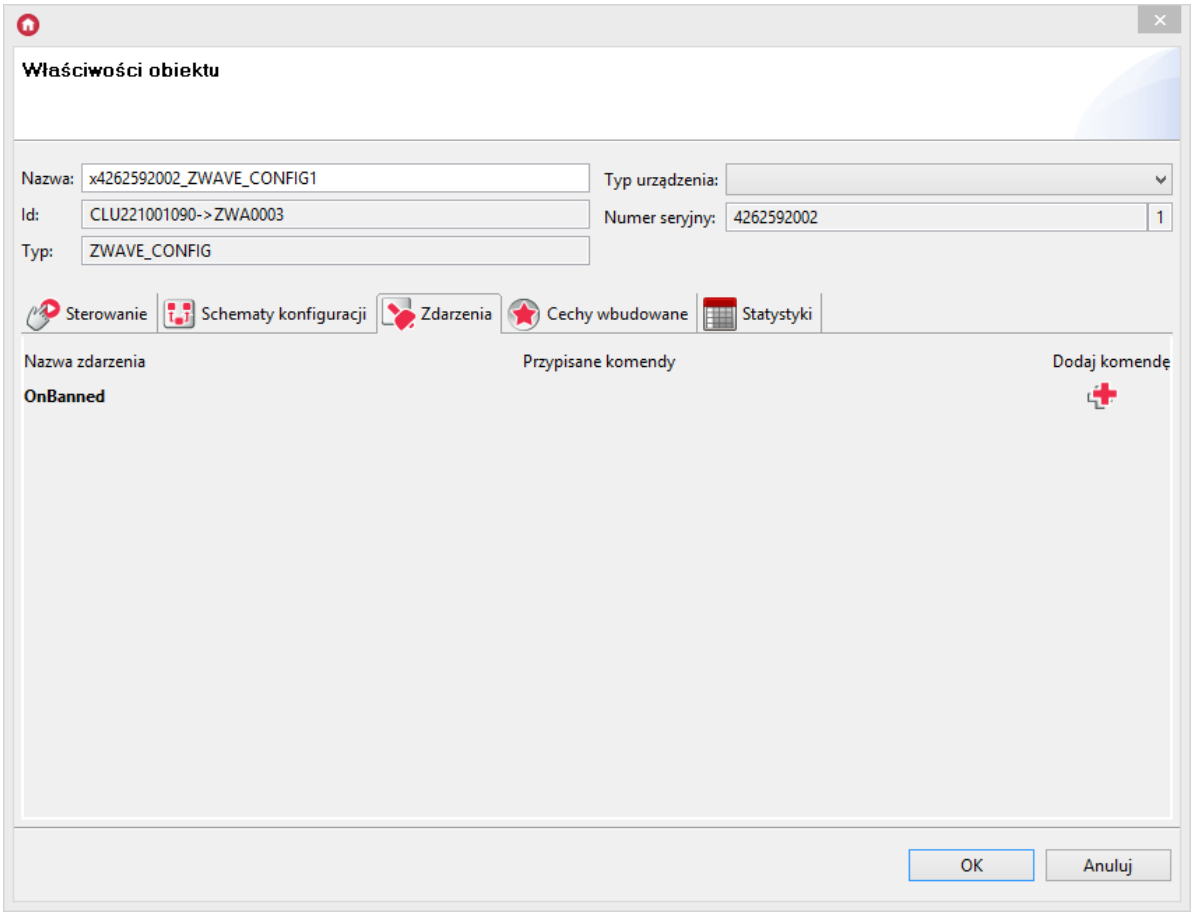


- Ponowna próba wysłania rozkazu do urządzenia ponawiania jest co 15 sekund - wykonywane są 3 próby komunikacji z urządzeniem.
- W przypadku 3 prób niepowodzenia komunikacji z modułem cecha `Banned` zostaje ustawiona na wartość 1 a cała komunikacja z modułem zostaje zablokowana.



Mechanizm blokowania komunikacji z modułem

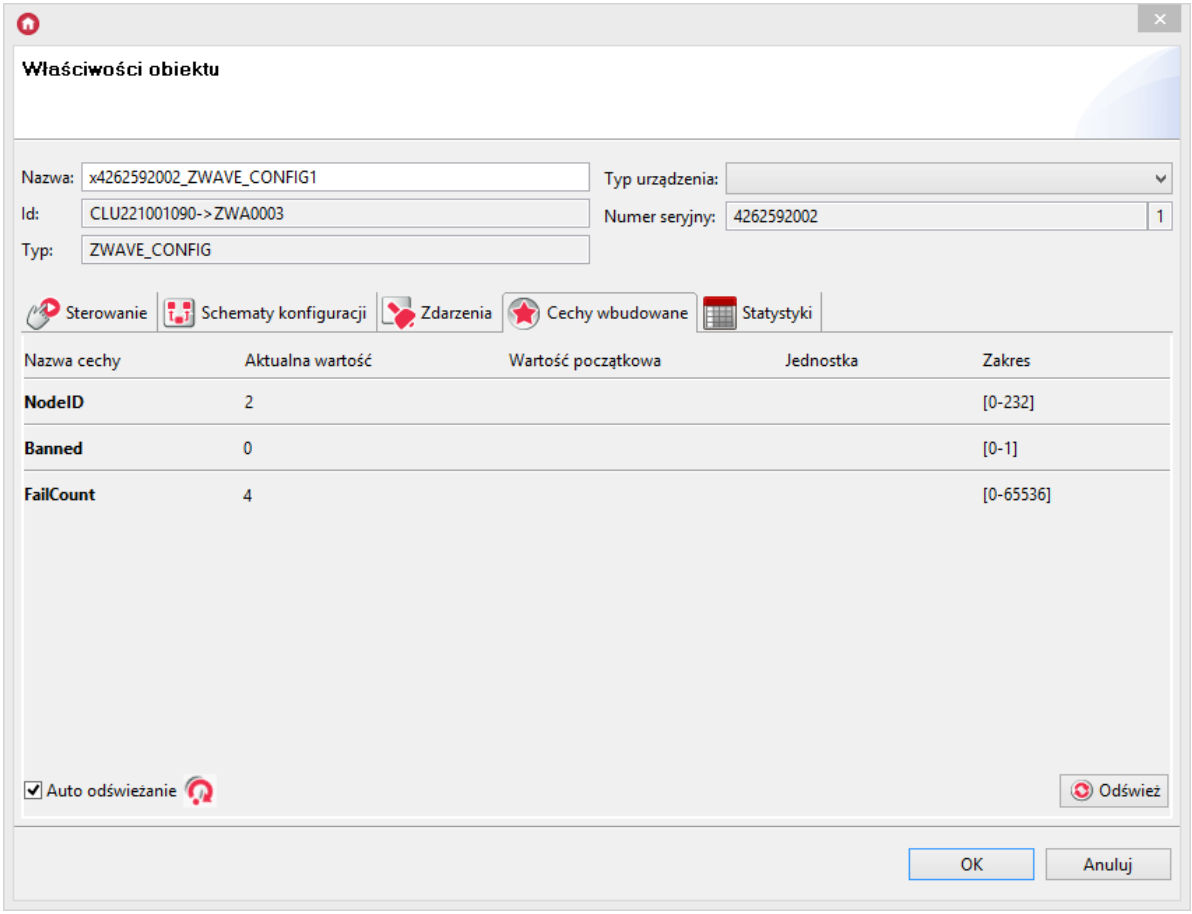
- W momencie ustawienia cechy `Banned` na wartość 1, następuje zablokowanie komunikacji z urządzeniem Z-Wave - oznacza to, że wszystkie wywołania akcji na urządzeniu (tj. zmiana stanu wyjścia, zapytanie o parametry) nie są wysyłane przez CLU do zablokowanego modułu.
- Istnieje możliwość przypisania dowolnej akcji w momencie zablokowania komunikacji z danym modułem, za pomocą zdarzenia `OnBanned`.



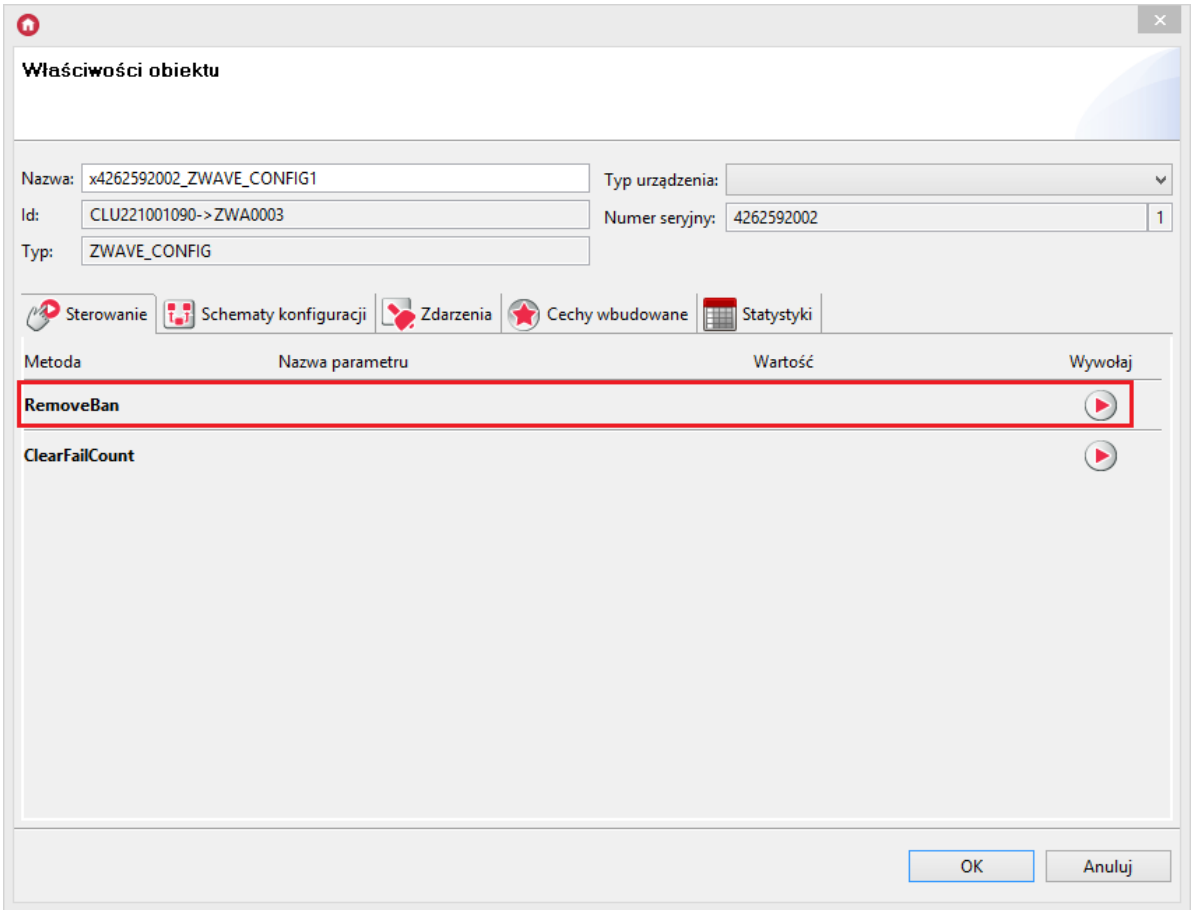
- Do zbanowanego modułu, co 1,5 minuty wysyłane jest krótkie zapytanie (NOP):
  - jeżeli moduł nie potwierdzi otrzymania zapytania, cecha `Banned` nadal przyjmuje wartość 1, a kolejne zapytanie zostaje ponawiane co 1,5 minuty,

> **Uwaga!** W przypadku, gdy więcej niż jeden moduł zostanie zbanowany, wtedy wysyłanie NOP odbywa się co 1,5 minuty do **kolejnego zbanowanego modułu**. Przykład:
>
> zbanowane 3 moduły A,B,C
>
> CLU - NOP -> moduł A
>
> 1,5 minuty przerwy
>
> CLU - NOP -> moduł B
>
> 1,5 minuty przerwy
>
> CLU - NOP -> moduł C
>
> 1,5 minuty przerwy
>
> CLU - NOP -> moduł A
>
> itd.

  - jeżeli moduł potwierdzi otrzymanie zapytania (ACK), cecha `Banned` zmienia wartość na 0 - oznacza to, że możliwe jest ponowne wysyłanie rozkazów do danego urządzenia.



- Istnieje możliwość ręcznego usunięcia blokady - za pomocą metody **RemoveBan**.



- Po wywołaniu tej metody cecha `Banned` zmienia wartość na 0 - oznacza to, że możliwe jest ponowne wysyłanie rozkazów do danego urządzenia.

> **Uwaga!** `RemoveBan` nie jest jednoznaczne z ponowną poprawną komunikacją z modułem – umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!

- W przypadku wystąpienia niepowodzenia komunikacji z modułem, cały mechanizm (zliczania niepowodzeń w komunikacji oraz blokowania) rozpoczyna się od początku.

Należy pamiętać, że w przypadku odblokowania komunikacji z modułem cecha `FailCount` nie jest zerowana - można to zrobić przy pomocy metody `ClearFailCount`.

6.4. Wskazówki konfiguracyjne sieci Z-Wave

Podczas tworzenia sieci Z-Wave istotne jest by:

- Konfiguracja sieci Z-Wave odbywała się po montażu urządzeń.

Z uwagi na fakt, że sieć Z-Wave definiowana jest statycznie, urządzenia Z-Wave należy linkować, gdy znajdują się w swoich docelowych lokalizacjach. Zmiana położenia urządzeń Z-Wave po ich dodaniu może powodować nieprzewidziane problemy z komunikacją w sieci Z-Wave - z wszystkimi urządzeniami!

- Antena (w modułach, które ją posiadają) nie była zwinięta ani zawinięta wokół modułu.

Antena powinna być umieszczona możliwie w kierunku „od modułu"

- Wybudzanie modułów bateryjnych nie następowało w tym samym czasie.

Wybudzanie modułów w tym samym czasie doprowadza do pojawienia się opóźnień w działaniu. W celu uniknięcia opisanej sytuacji, należy dla wszystkich urządzeń stosować inne czasy wybudzania (w obiekcie ZWAVE_WAKEUP dla modułów bateryjnych) i dobierać je w taki sposób, by ustawione czasy miały możliwie jak największą „najmniejszą wspólną wielokrotność", np.: 57min, 58min, 59min, 60min, 61min, itd...

- W sieci Z-Wave nie znajdowały się nieaktywne moduły (uszkodzone lub niepoprawnie usunięte).

Podlinkowany moduł, którego brakuje w systemie powoduje ciągłe próby odnowienia komunikacji z nim, co z kolei może wprowadzać chwilowe opóźnienia i braki komunikacji także z innymi urządzeniami.

> **Uwaga!** Dla CLU Z-Wave umieszczonego w skrzynce/szafie zaleca się wykorzystanie dłuższych anten oraz wyniesienie ich poza rozdzielnię.

6.5. Czyszczenie informacji o węzłach

Istnieje możliwość jednoczesnego usunięcia wszystkich modułów Z-Wave z CLU. Do tego celu służy funkcja *HardReset* - patrz pkt XI.1..

7. Wysyłanie konfiguracji do CLU



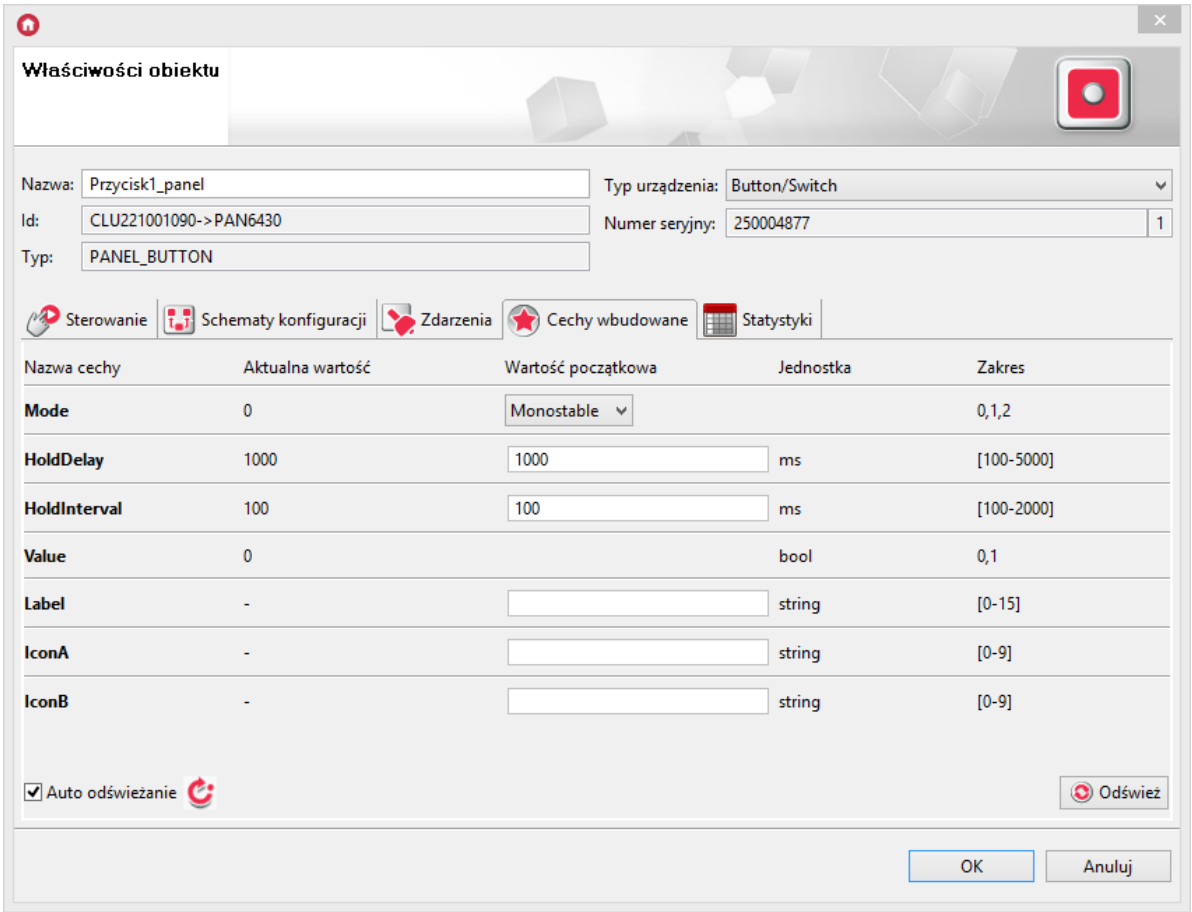
Konfiguracja przechowywana jest w OM i dopóki nie zostanie przesłana do CLU, nie jest brana pod uwagę w działaniu systemu. Żeby wysłać konfigurację do CLU należy nacisnąć przycisk `Wyślij konfigurację` w menu .

Object Manager wykryje, na których CLU została dokonana zmiana i wyśle konfigurację.

> **Uwaga!** Po wysłaniu konfiguracji CLU zostaną zrestartowane, przez co lampy podłączone do systemu mogą zgasnąć, a system może przez kilka sekund nie reagować na naciśnięcia wyłączników itp.

8. Wartości początkowe cech

Każdy obiekt w systemie posiada swoją własną listę cech, z których część może być ustawiana. Cechy mogą być ustawiane w trakcie startu systemu (restart CLU), dzięki czemu możliwa jest jednorazowa konfiguracja zachowania obiektów (np. ustawienie przycisków panelu dotykowego jako bistabilne, monostabilne). Wartości początkowe cech ustawiane są w zakładce: *Cechy wbudowane* w formatce obiektu (CLU, wejścia, wyjścia):



By ustawić wybraną cechę, należy w odpowiednim polu, w kolumnie `Wartość początkowa` wpisać żądaną wartość, a następnie wysłać konfigurację do CLU.

9. Tworzenie podstawowych powiązań

Wywoływanie reakcji w systemie (np. załączenie oświetlenia po wciśnięciu klawisza) jest realizowane przez tworzenie powiązań pomiędzy obiektami. Z reguły są to powiązania pomiędzy wejściem (np. wyłącznik) a wyjściem (lampa). System nie ogranicza jednak w żaden sposób tworzenia powiązań i umożliwia ich tworzenie pomiędzy zdarzeniami wyjść ze zdarzeniami dowolnych innych obiektów, przez co możliwe jest, np. włączenie oświetlenia LED-owego w momencie wyłączenia lampy głównej.

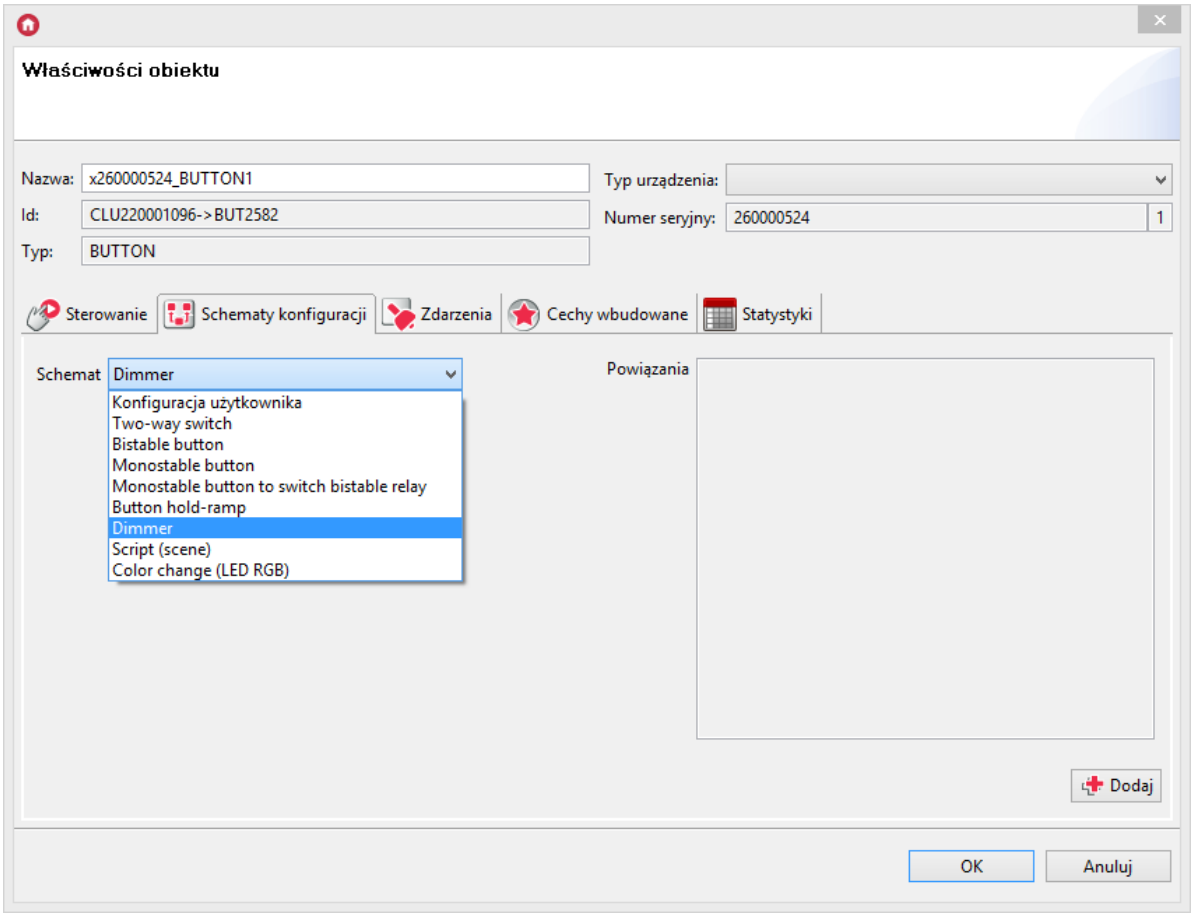
Powiązania mogą być tworzone na dwa sposoby:

- Poprzez użycie schematów konfiguracji - pozwala to na szybkie tworzenie typowych powiązań typu wyłącznik-lampa;
- Poprzez ręczne tworzenie powiązań zdarzenie-metoda - co zapewni ogromną elastyczność w tworzeniu logiki systemu.

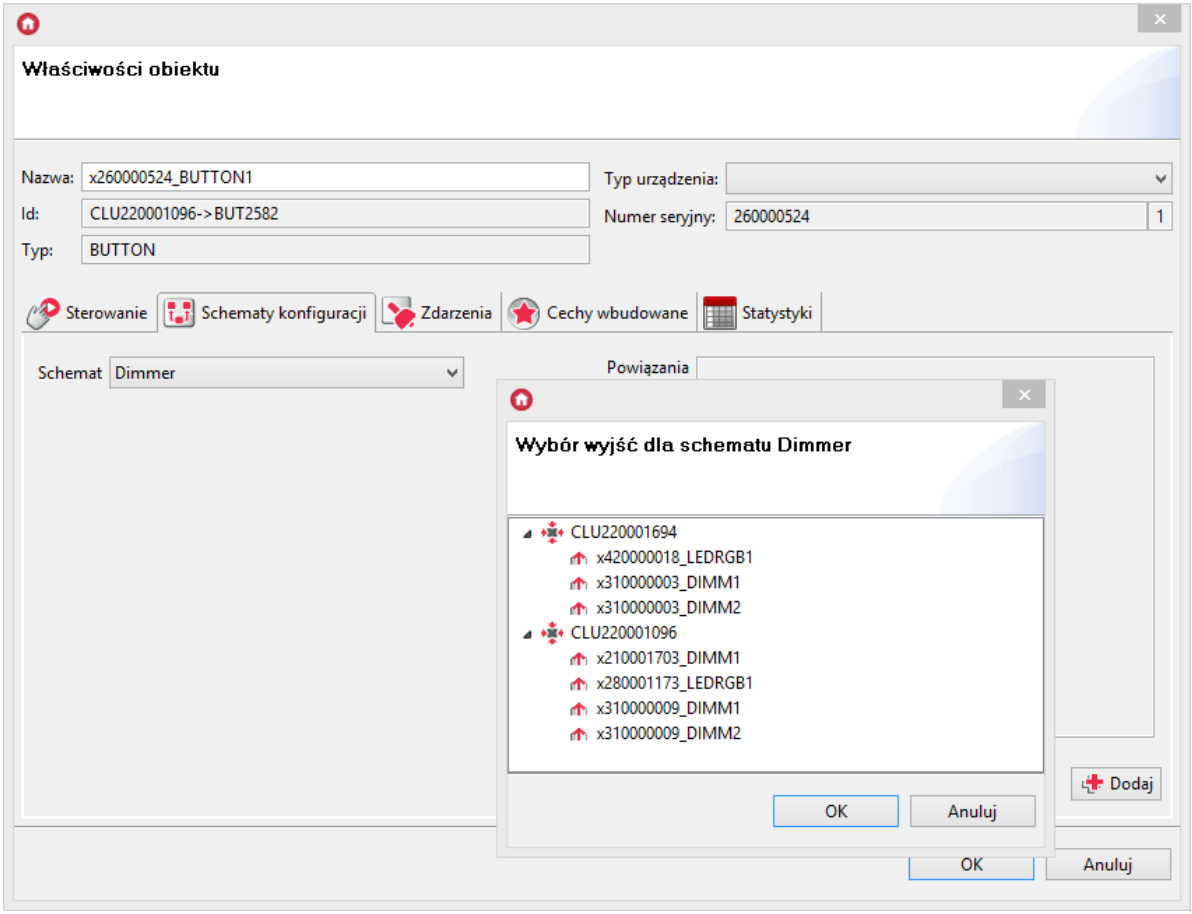
Tworzenie powiązań poprzez użycie schematów konfiguracji

W celu utworzenia powiązania przy pomocy schematu konfiguracji należy wykonać następujące czynności:

- Dwukrotnie kliknąć na wejście, które ma zostać powiązać z wyjściem;
- Przejść do zakładki *Schematy konfiguracji*, z listy wybrać interesujący schemat;



- Klikając Dodaj wybrać wyjścia, które mają być wyzwalane;



W polu wyboru wyjść dla schematu pojawią się wyłącznie wyjścia, dla których możliwe jest przypisanie danej logiki.

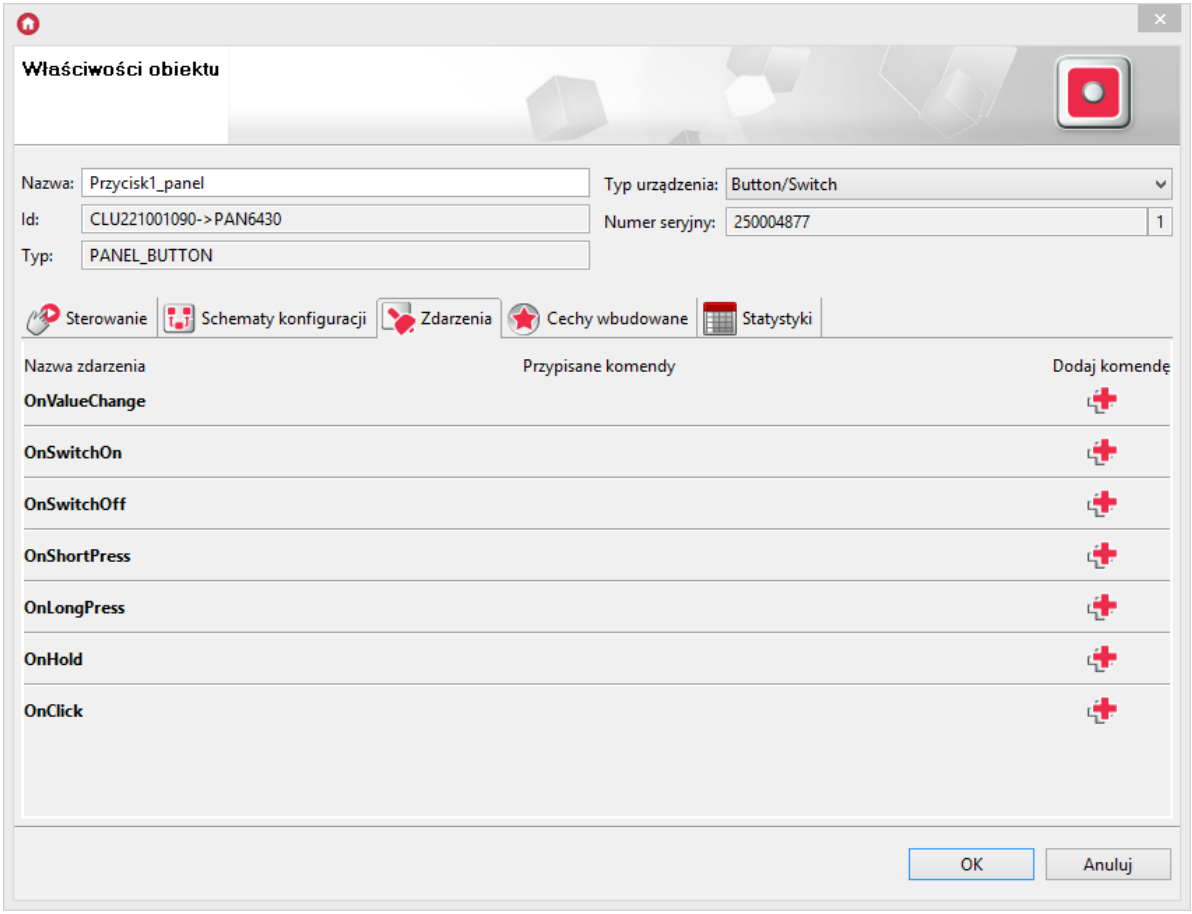
Aby wybrać więcej niż jedno wyjście należy zaznaczać kolejne obiekty przytrzymując na klawiaturze klawisz `Ctrl`, lub zaznaczyć przedział obiektów, od pierwszego do ostatniego, przytrzymując klawisz `Shift`. Po zatwierdzeniu wybranych wyjść dla schematu, Object Manager automatycznie utworzy powiązania odpowiednich zdarzeń z metodami powiązanych obiektów.

- Skonfigurować pozostałe wejścia i wysłać konfigurację do CLU.

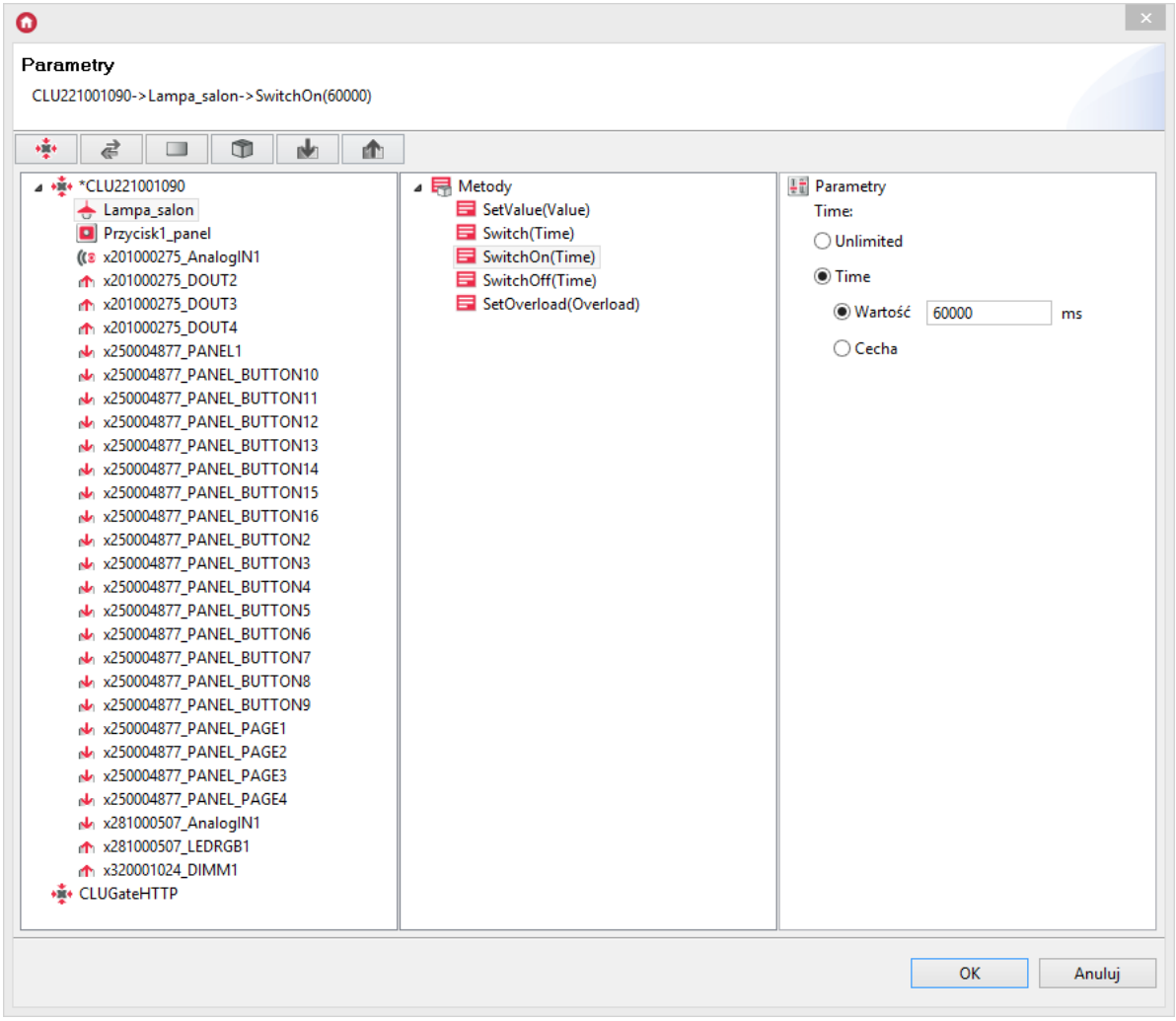
Ręczne tworzenie powiązań zdarzenie-metoda

By ręcznie utworzyć powiązanie zdarzenie-metoda należy:

- Z listy obiektów w systemie wybrać interesujący obiekt i kliknąć na niego dwukrotnie;
- Przejść do zakładki `Zdarzenia`:

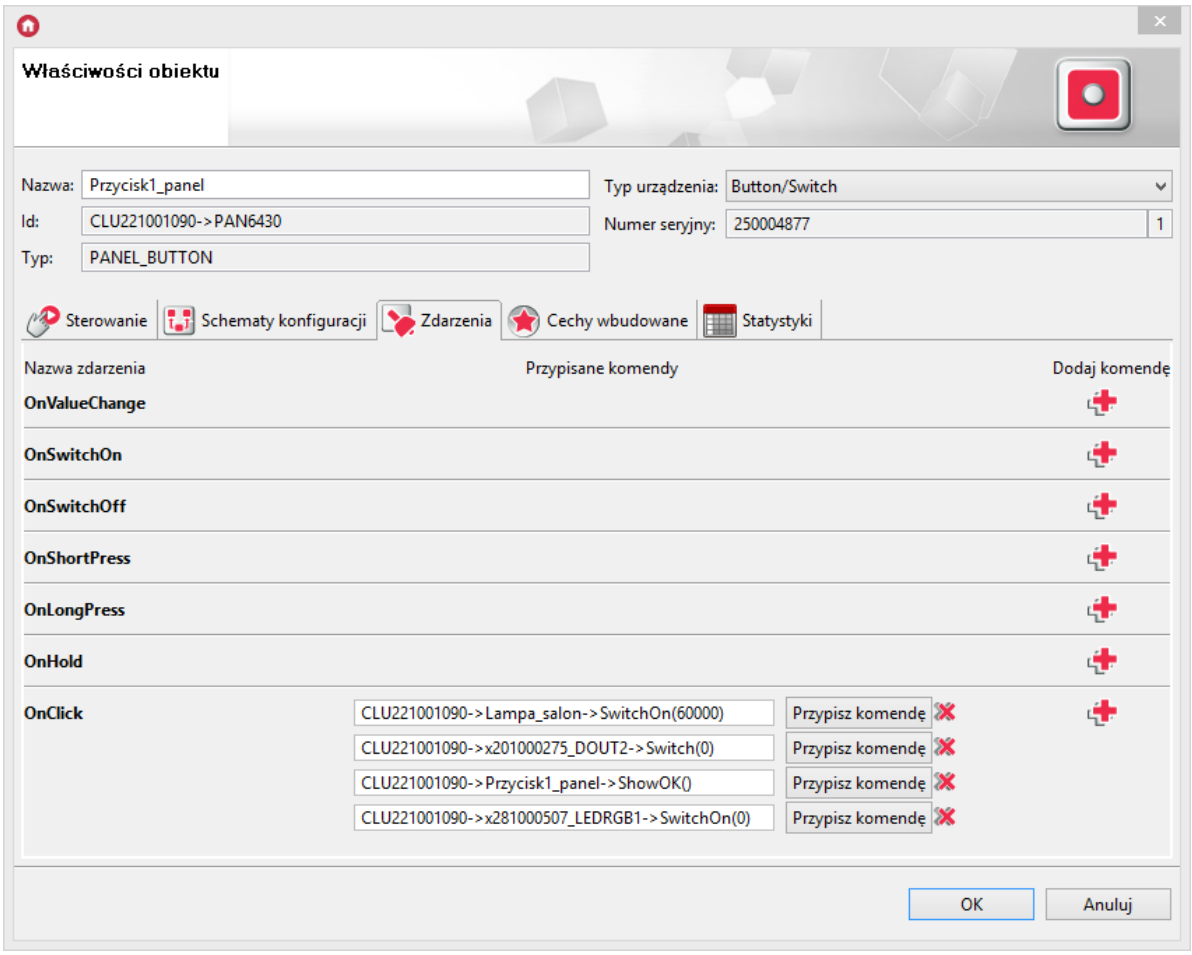


- Znaleźć na liście zdarzenie, które ma zostać powiązane i kliknąć `+`;
- W formatce wybierania metody wybrać kolejno obiekt, metodę i parametry:

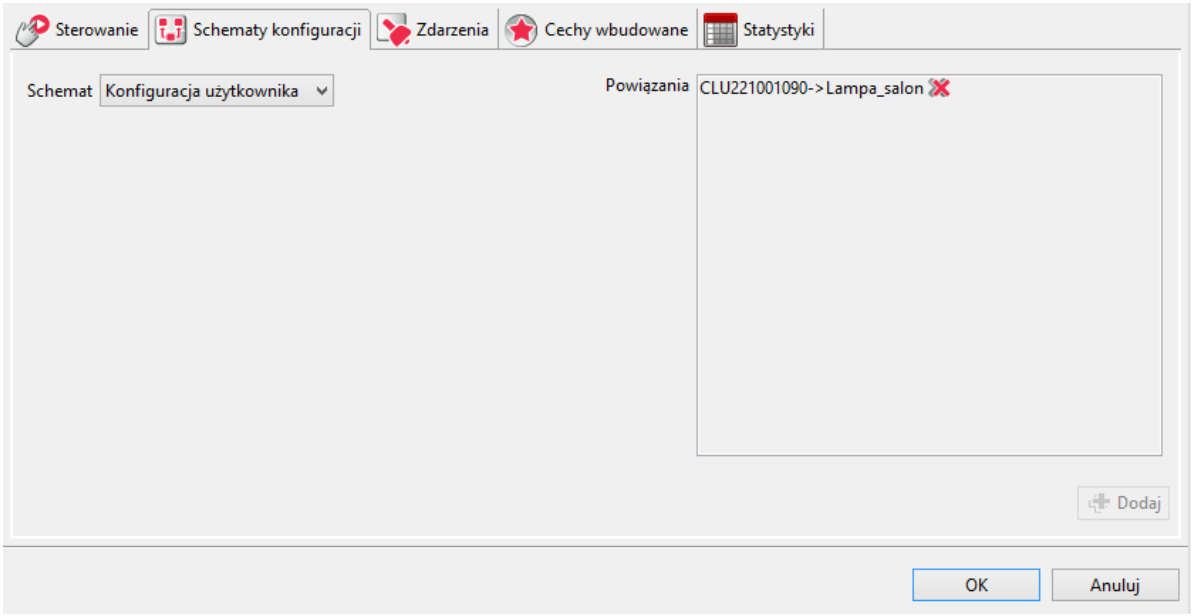


- Skonfigurować pozostałe zdarzenia i wysłać konfigurację do CLU.

Do każdego zdarzenia można dodać do 4 metod wyjść. Jeśli konieczne jest dodanie większej liczby metod lub warunków, proponuje się utworzenie skryptu.



W przypadku, gdy użytkownik stworzył własne powiązania zdarzenie-metoda, korzystając z zakładki `Zdarzenia`, widoczne są one na liście jako `Konfiguracja użytkownika`.



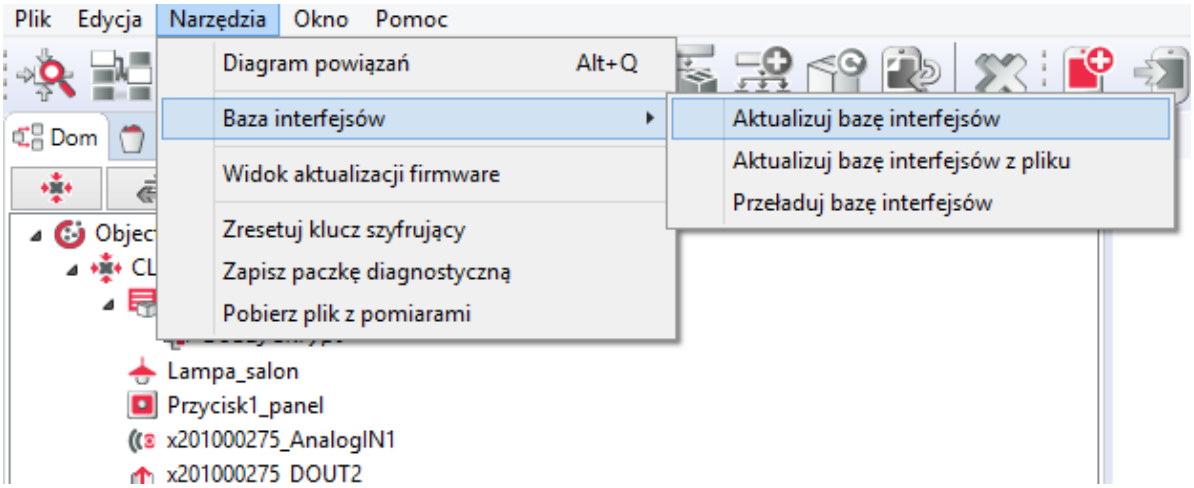
10. Przeprowadzanie aktualizacji

10.1. Proces aktualizacji bazy interfejsów

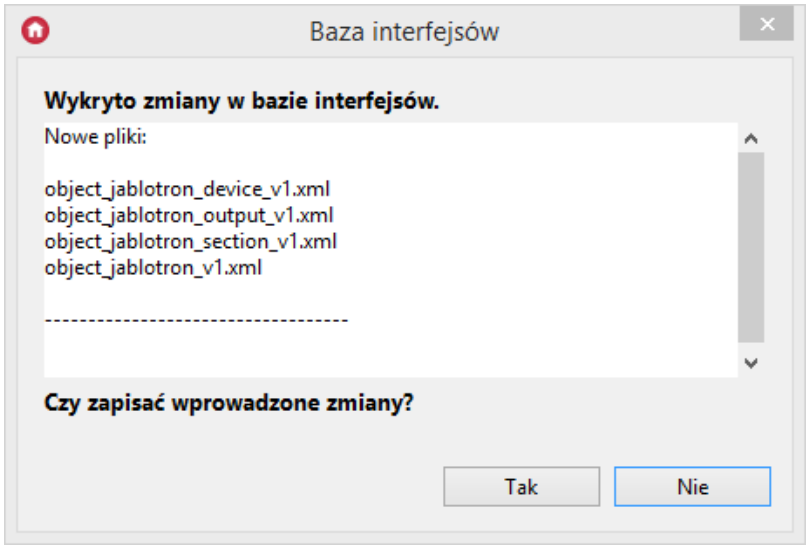
Jeśli przy pierwszym uruchomieniu Object Managera zaznaczona została opcja *automatycznego aktualizowania bazy interfejsów*, nie ma konieczności jej dodatkowego uruchamiania. W przeciwnym wypadku, należy pamiętać o regularnym przeprowadzaniu aktualizacji. Aktualizacja bazy interfejsów powinna zostać wykonana zawsze przed aktualizacją oprogramowania danego modułu Grenton, a do jej przeprowadzenia niezbędne jest połączenie z internetem (aktualizacja odbywa się z serwera).

W celu zaktualizowania bazy interfejsów w Object Managerze należy:

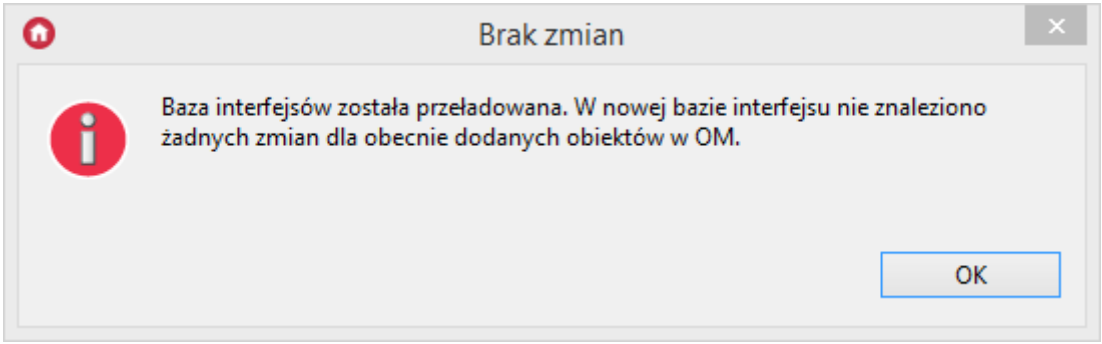
- Z paska menu wybrać `Narzędzia`.
- Zaznaczyć pozycję *Baza interfejsów*.
- Z wyświetlonej listy wybrać *Aktualizuj bazę interfejsów*:



- Po chwili pojawi się okno z wykrytymi zmianami w bazie interfejsów, które należy zaakceptować, klikając przycisk *OK*:

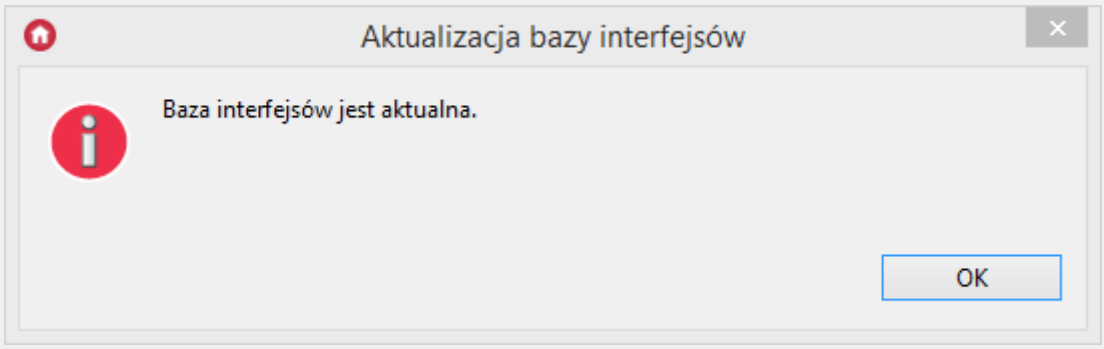


- Następnie wyświetlone zostanie okno informujące o zakończeniu przeładowywania bazy interfejsów:



- Ostatnim etapem jest przesłanie konfiguracji do jednostki centralnej, które następuje automatycznie.

Uwaga! Jeżeli konfiguracja jest aktualna, to po wyborze opcji: *Aktualizuj bazę interfejsów* wyświetlony zostanie komunikat:



10.2. Proces aktualizacji firmware'u CLU

Aktualizację firmware'u przeprowadza się w celu dodania obsługi nowych urządzeń oraz zwiększenia możliwości systemu. Więcej szczegółów można znaleźć w Release Notes.

Uwaga! Aktualizacja firmware CLU 2.0 możliwa jest tylko w Object Managerze w wersji 1.3.0.1927 lub wyższej!

Uwaga!

Wyświetlanie statusu urządzeń dostępne jest w Object Managerze w wersji 1.3.5.240201 lub wyższej!

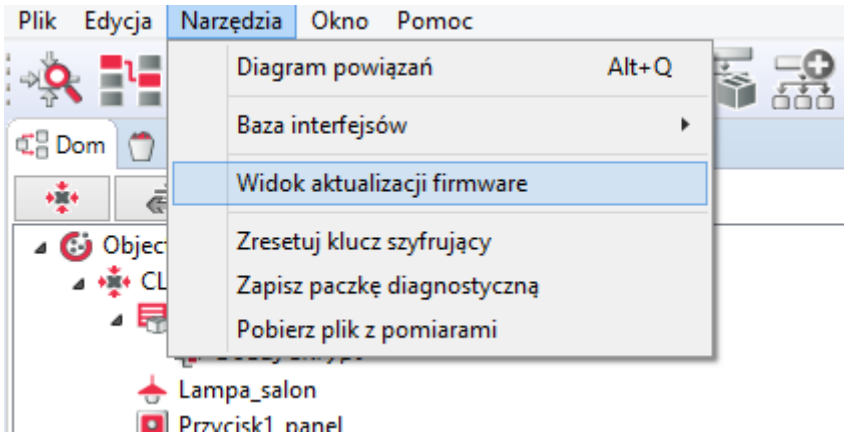
Uwaga! Aby proces aktualizacji przebiegł poprawnie należy spełnić poniższe wymagania:

- Komputer z OM musi być podłączony do zasilacza, nie może być na zasilaniu bateryjnym.
- Połączenie sieciowe między CLU<->router<->komputer musi być kablowe, nie można wykorzystywać połączenia WIFI.
- Podczas upgrade'u firmware'u nie należy wykonywać żadnych działań z systemem Grenton.

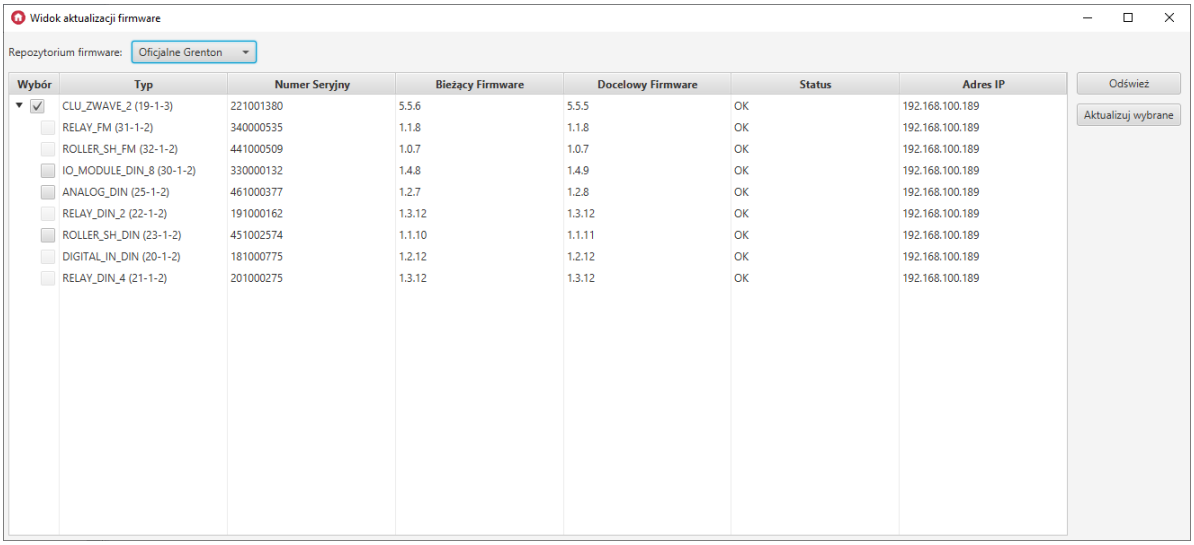
A. Aktualizacja z serwera Grenton

Chcąc zaktualizować firmware na CLU należy:

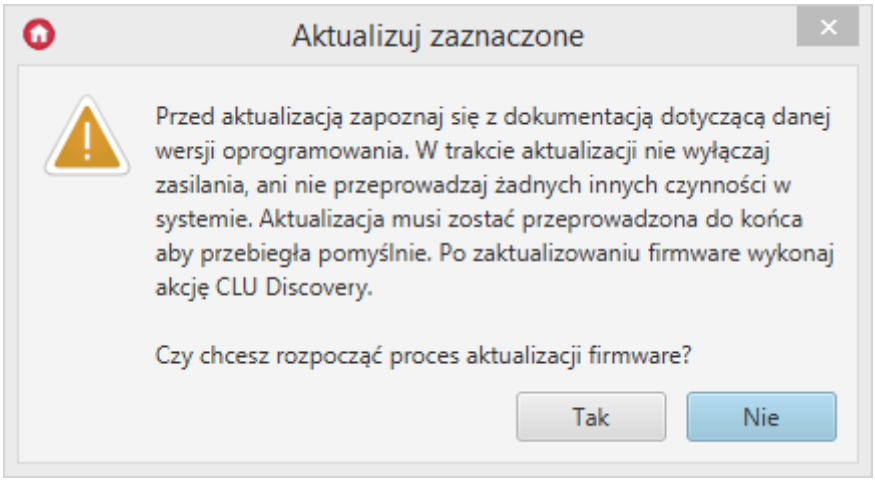
- Z paska menu wybrać `Narzędzia`.
- Wybrać opcję *Widok aktualizacji firmware*:



- Zaznaczyć obiekt typu `CLU_ZWAVE_2`. Zaznaczenie pola wyboru jest możliwe tylko w przypadku, gdy bieżący firmware na CLU jest nieaktualny:

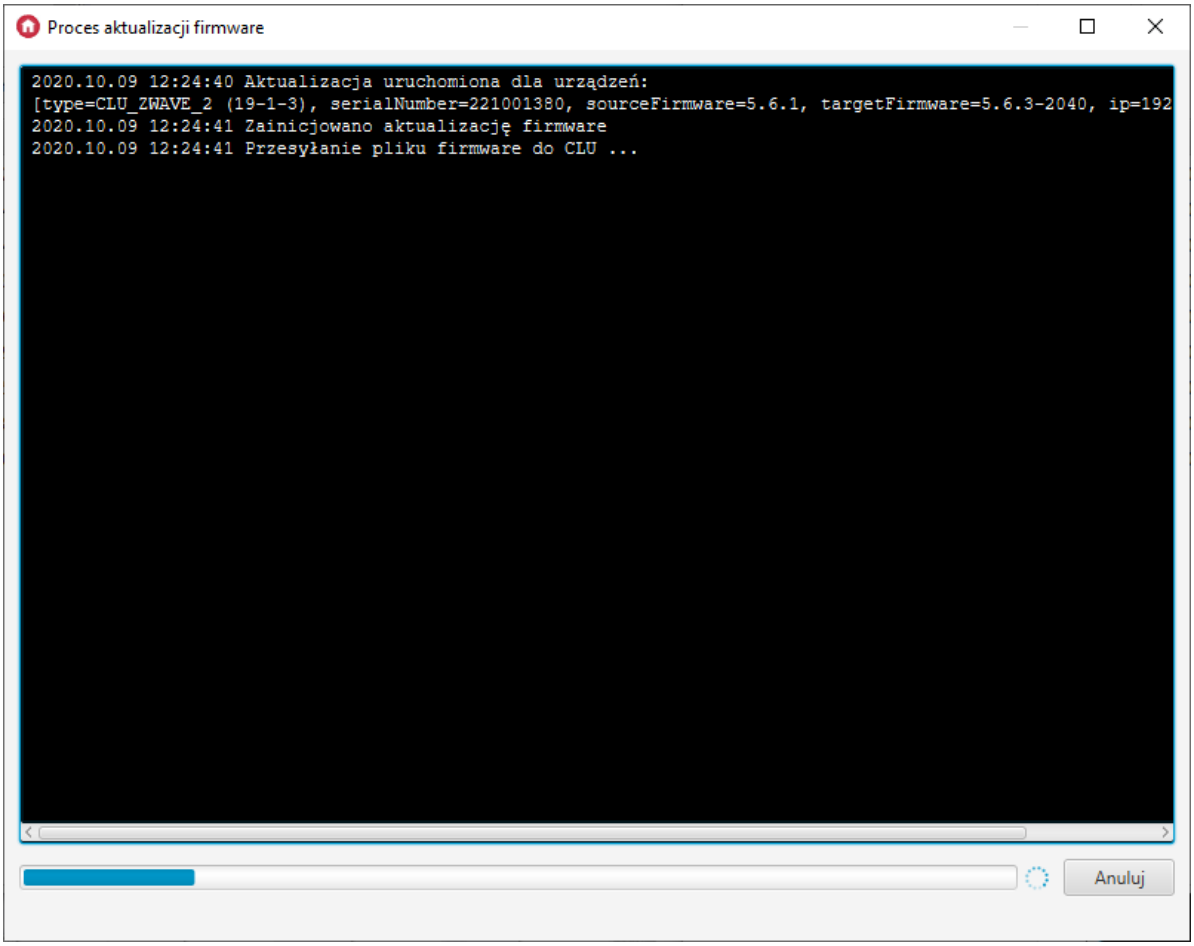


- Wybrać opcję `Aktualizuj wybrane`. Po zapoznaniu się z komunikatem, wybrać „Tak" aby kontynuować:

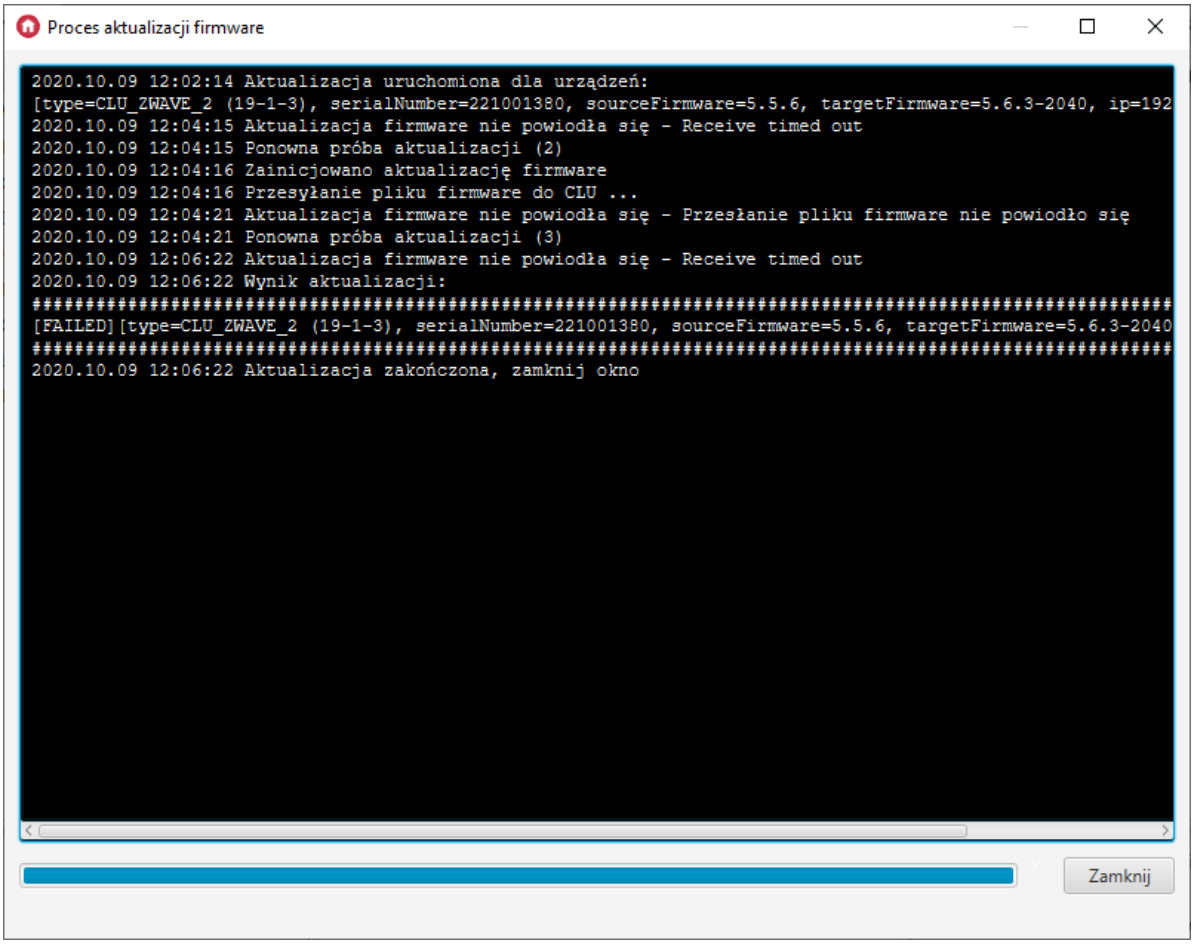


- Po zaakceptowaniu rozpocznie się proces aktualizacji:

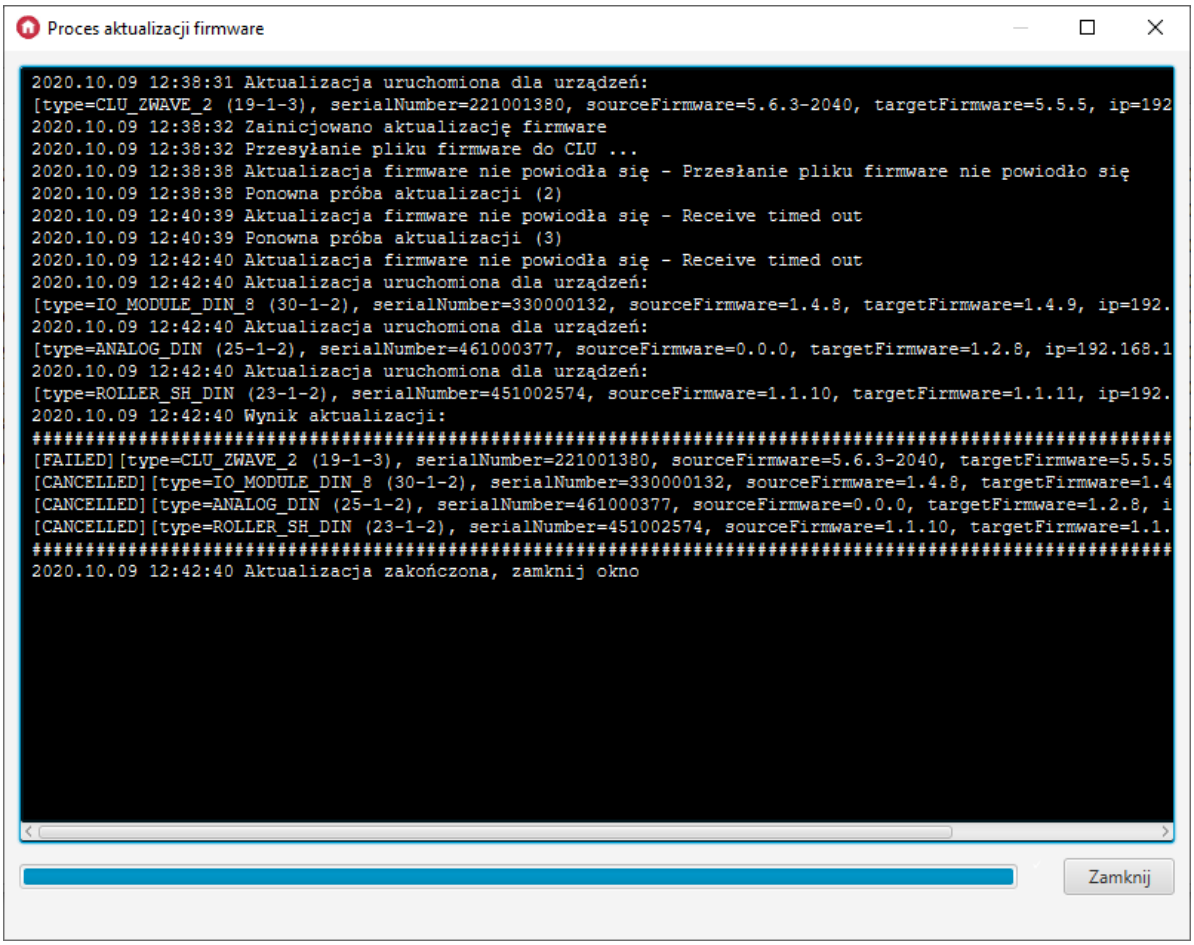
Uwaga! Podczas procesu aktualizacji nie wolno wyłączać zasilania ani przeprowadzać innych czynności w systemie. Aby aktualizacja przebiegła pomyślnie należy przeprowadzić ją do końca.



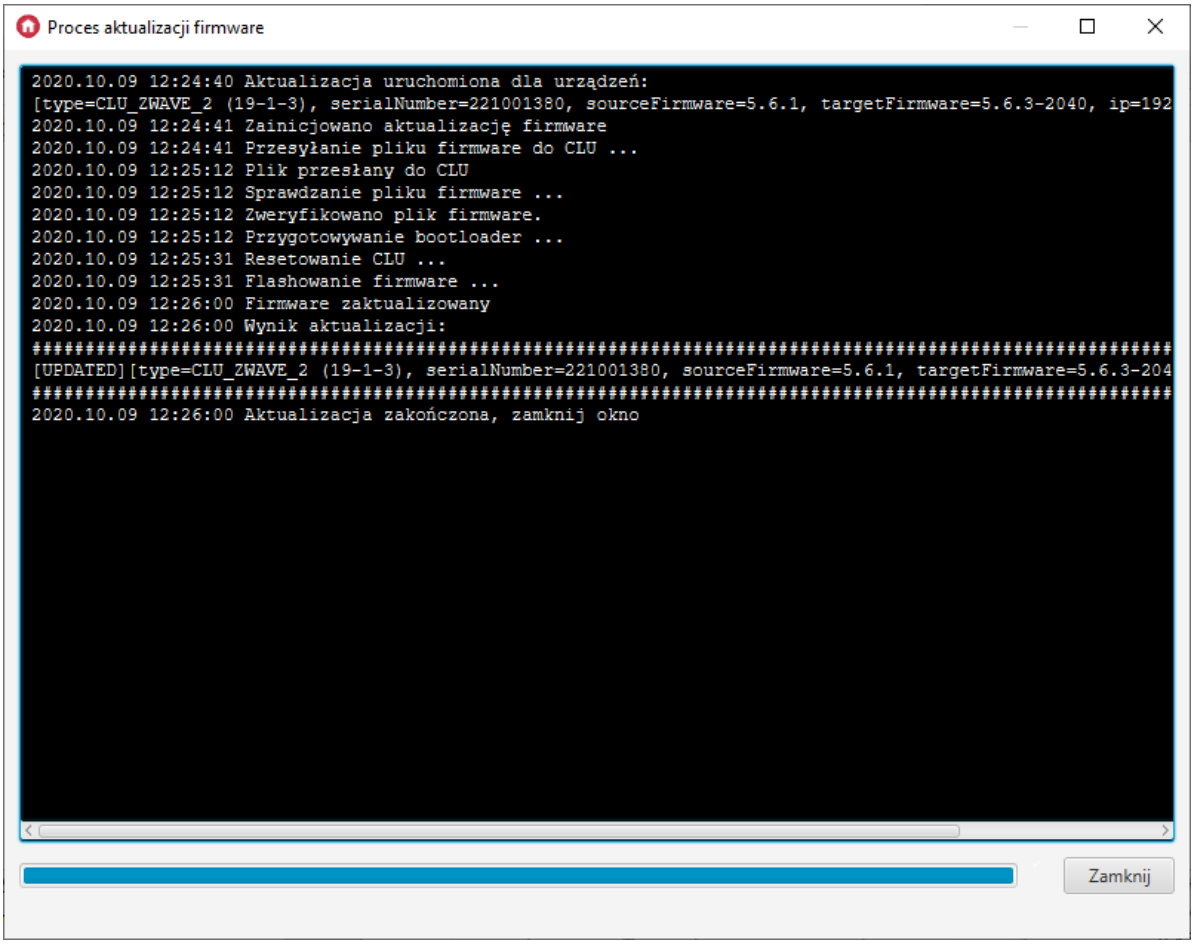
- W przypadku niepowodzenia, nastąpią dwie ponowne próby aktualizacji modułu. Jeśli również zakończą się niepowodzeniem, przy module pojawi się informacja [FAILED]:



- Jeśli aktualizacja CLU zakończy się niepowodzeniem, a w kolejce do aktualizacji będą moduły TF-Bus, ich aktualizacja zostanie anulowana [CANCELLED]:



- W przypadku poprawnej aktualizacji, przy zaktualizowanym module pojawi się informacja [UPDATED]:



- Aby zakończyć proces aktualizacji należy zamknąć okno - klikając przycisk `Zamknij`.

Jeżeli aktualizacja przebiegła pomyślnie, bieżąca wersja firmware'u po odświeżeniu powinna być zgodna z docelową wersją, a status urządzenia powinien wskazywać "OK". Jeśli OM nie może nawiązać połączenia z CLU, wówczas pojawi się status "DISCONNECTED".

> **Uwaga!** Po zakończeniu aktualizacji należy wykonać CLU Discovery.

Istnieje możliwość aktualizacji większej ilości CLU w ramach jednego procesu. W tym celu na liście wyboru należy zaznaczyć wszystkie CLU, które mają zostać zaktualizowane.

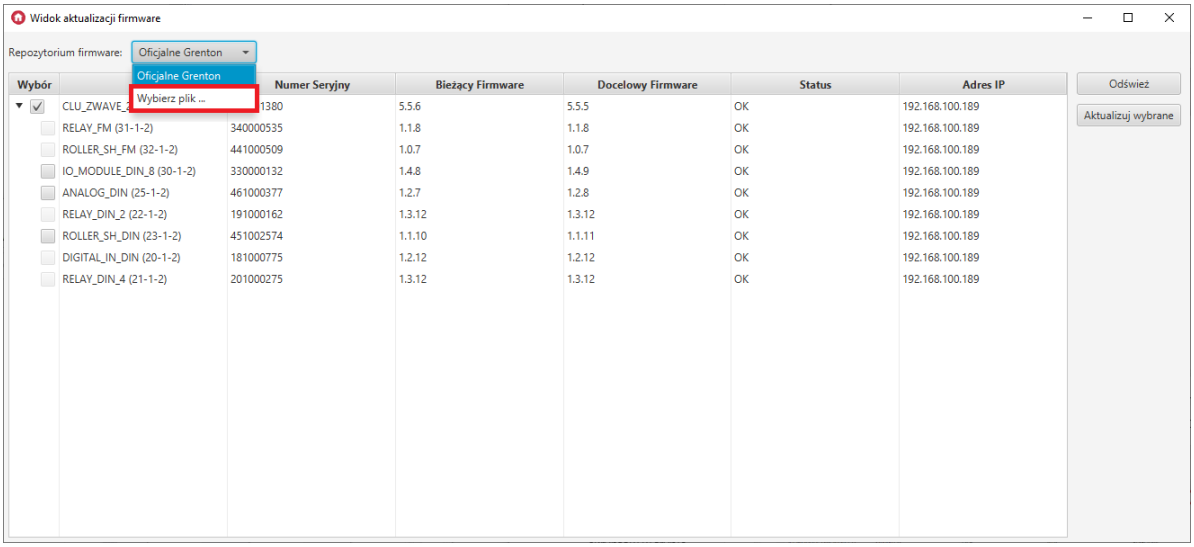
B. Aktualizacja za pomocą pliku .ZIP

Domyślnie informacje o aktualnym firmware pobierane są z serwera Grenton. Można jednak dokonać aktualizacji z pliku. Aktualizacja z pliku odbywa się za pomocą paczek .zip przygotowanych przez Grenton.

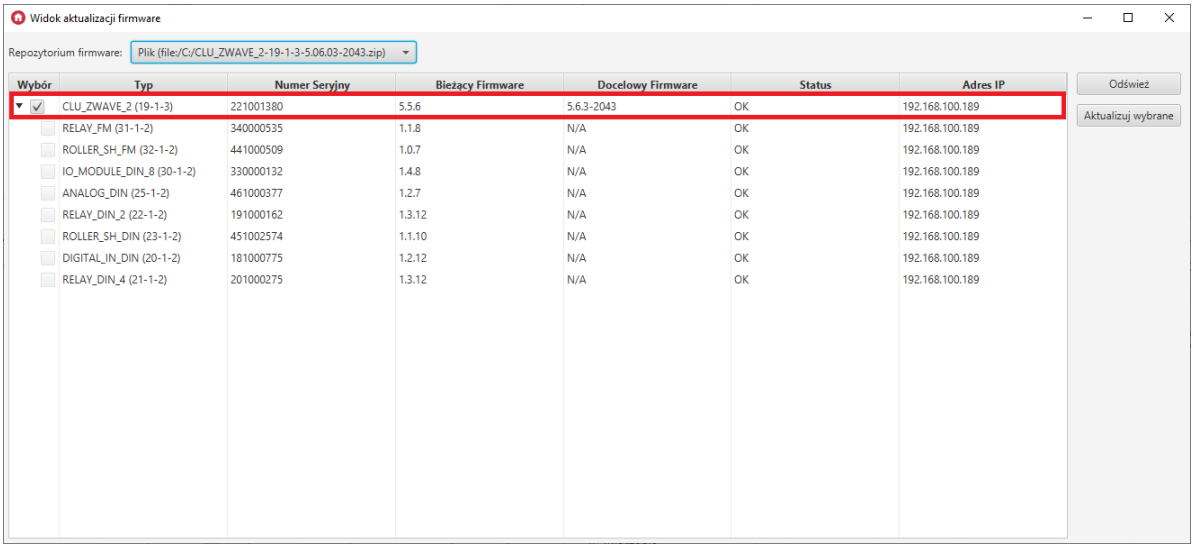
> **Uwaga!** Nie należy zmieniać nazwy pliku .zip dostarczonego przez Grenton. Plik musi posiadać odpowiednią nazwę, aby został prawidłowo wczytany.

Aby dokonać aktualizacji z pliku należy:

- W pozycji *Repozytorium firmware* wybrać opcję `Wybierz plik`:



Po załadowaniu pliku nastąpi możliwość zaznaczenia modułu, który można zaktualizować. W pozycji *Docelowy firmware* pojawi się numer wersji firmware, do którego nastąpi aktualizacja.



- Po zaznaczeniu modułu wybrać opcję `Aktualizuj wybrane` i kontynuować instalację, analogicznie jak w przypadku standardowej aktualizacji z serwera patrz pkt VI.10.2.A..

10.3. Proces aktualizacji firmware'u modułów z serii 2.0

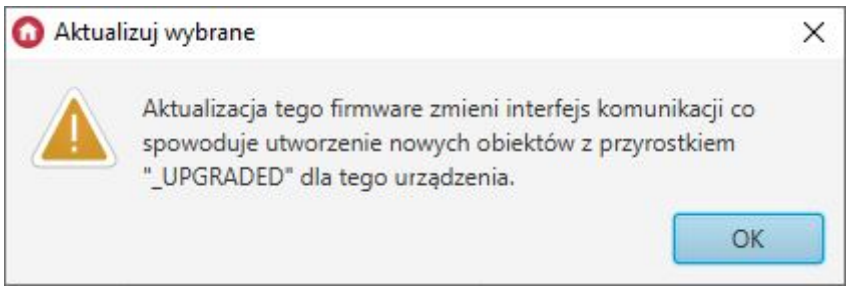
Uwaga! Proces aktualizacji firmware urządzeń możliwy jest tylko dla modułów z serii 2.0 !

Uwaga! Aby proces aktualizacji urządzeń TF-Bus przebiegł poprawnie należy spełnić poniższe wymagania:

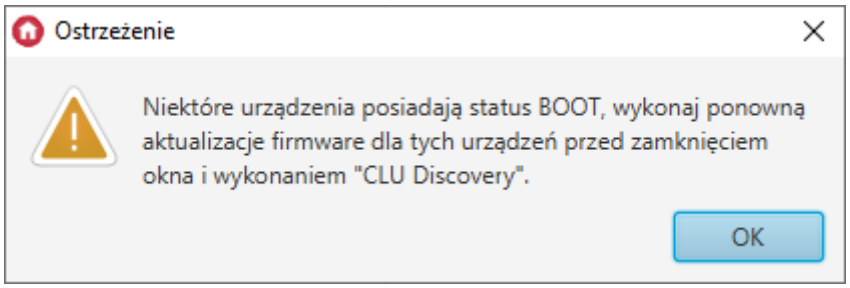
- Komputer z OM musi być podłączony do zasilacza, nie może być na zasilaniu bateryjnym.
- Połączenie sieciowe między CLU<->router<->komputer musi być kablowe, nie można wykorzystywać połączenia WIFI.
- Podczas upgrade'u firmware'u nie należy wykonywać żadnych działań z systemem Grenton.
- Zaczynamy od upgrade samego firmware CLU, potem Discovery i w kolejnym kroku można wykonać -> upgrade modułów, po którym należy również wykonać Discovery.

Aktualizacja modułów z serii 2.0 przebiega podobnie jak w przypadku aktualizacji firmware CLU. Przed rozpoczęciem aktualizacji należy pamiętać o następujących kwestiach:

- Aktualizacja firmware danego modułu jest możliwa tylko w przypadku, gdy firmware na CLU jest aktualny. W innym przypadku należy również zaznaczyć CLU, które zostanie zaktualizowane w pierwszej kolejności.
- Aktualizacja odbywa się dla wszystkich modułów tego samego typu. Zaznaczając dany moduł, zostają zaznaczone wszystkie moduły tego samego typu znajdujące się na liście (o ile występują).
- Jeżeli dla danego modułu zostanie wykonana aktualizacja do wersji zmieniającej `firmwareApiVersion` modułu (przykładowo z wersji **1**.x.x do wersji **2**.x.x), w kolumnie `Docelowy firmware` przy danym module wyświetlana jest ikona ⚠, natomiast po zaznaczeniu modułu wyświetlany jest ostrzeżenie o zmianie interfejsu oraz o utworzeniu nowych obiektów (`_UPGRADED`) dla urządzenia po wykonaniu CLU Discovery.



- W momencie rozpoczęcia procesu, nie ma możliwości zatrzymania aktualizacji dla urządzenia które w danym momencie jest aktualizowane. Przerwanie aktualizacji nastąpi po zakończeniu procesu dla danej grupy urządzeń (anulowana dla kodlejnej grupy modułów).
- W niektórych przypadkach, aktualizacja danego urządzenia może przebiegać wieloetapowo. W takim przypadku po zakończeniu procesu aktualizacji należy sprawdzić, czy nie jest dostępna kolejna nowa wersja firmware'u dla danego modułu.
- Po przeprowadzeniu aktualizacji modułu należy sprawdzić czy bieżąca wersja firmware'u po odświeżeniu jest zgodna z docelową wersją, oraz czy status urządzenia wskazuje wartość "OK".
- Jeżeli po aktualizacji dany moduł wskazuje wartość statusu "BOOT", oznacza to, że proces aktualizacji oprogramowania został przerwany i urządzenie nadal oczekuje na nowy firmware. Po zamknięciu okna procesu aktualizacji pojawi się ostrzeżenie:



  W takiej sytuacji należy ponowić aktualizację dla danego modułu. Jeżeli posiada on status "DETACHED" oznacza to, że CLU nie może nawiązać połączenia z danym modułem. W takiej sytuacji należy sprawdzić połączenia magistrali TF-Bus i wykonać restart napięciowy całego systemu.

> **Uwaga!** Po prawidłowym zakończeniu aktualizacji należy wykonać CLU Discovery. Nie należy wykonywać CLU Discovery w przypadku, gdy jakikolwiek moduł posiada status "BOOT" bądź "DETACHED"!

10.4. Status CLU / modułów w oknie aktualizacji firmware'u

Bieżący status modułów wyświetlany jest w tabeli widoku aktualizacji firmware. Jeśli podczas otwartego okna aktualizacji nastąpi zmiana statusu, konieczne jest odświeżenie listy za pomocą przycisku "Odśwież".

Uwaga!

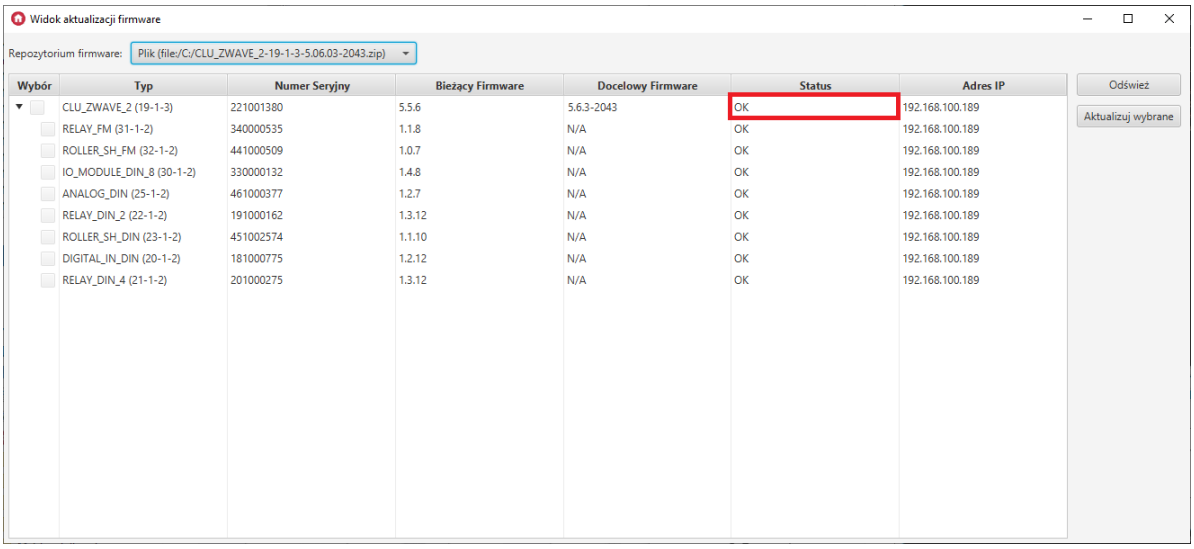
Wyświetlanie statusu urządzeń dostępne jest w Object Managerze w wersji 1.3.5.240201 lub wyższej!

Uwaga!

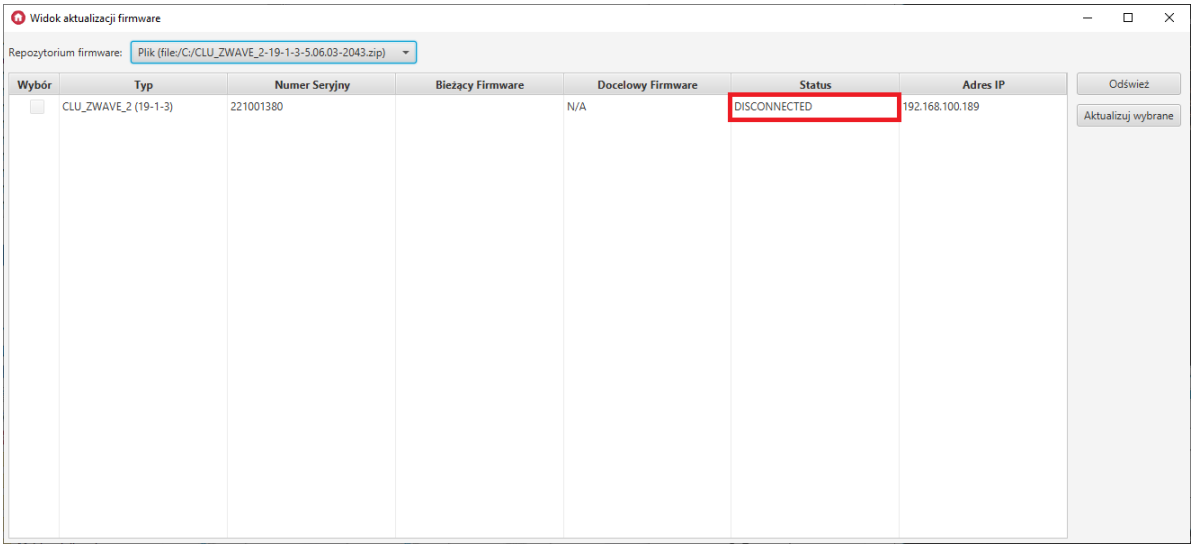
Obsługa statusów urządzeń dostępna jest dla CLUZ wersja fw. 5.06.03-2043 lub wyższa!

A. Status CLU

Status: OK - Prawidłowe połączenie OM z CLU.

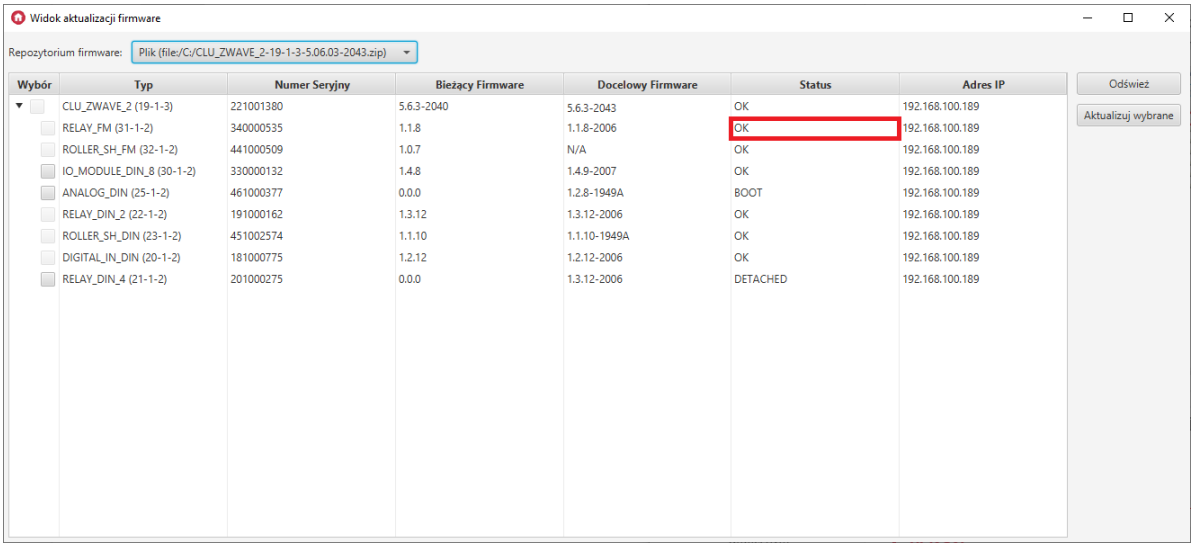


Status: DISCONNECTED - OM nie może połączyć się z CLU. Sytuacja pojawia się, gdy OM nie dostaje odpowiedzi od CLU. W takiej sytuacji należy sprawdzić, czy przewód sieciowy jest prawidłowo połączony do CLU / routera / switcha lub wykonać reset napięciowy CLU.



B. Status modułów TF-Bus

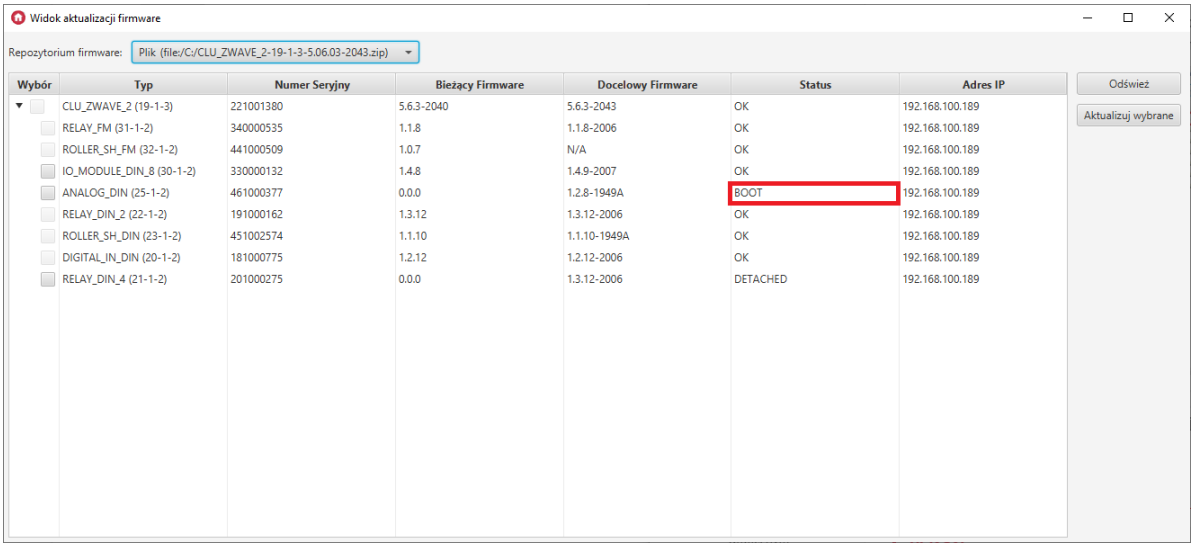
Status: OK - Prawidłowe połączenie modułu z CLU.



Wybór	Typ	Numer Seryjny	Bieżący Firmware	Docelowy Firmware	Status	Adres IP
	CLU_ZWAVE_2 (19-1-3)	221001380	5.6.3-2040	5.6.3-2043	OK	192.168.100.189
	RELAY_FM (31-1-2)	340000535	1.1.8	1.1.8-2006	OK	192.168.100.189
	ROLLER_SH_FM (32-1-2)	441000509	1.0.7	N/A	OK	192.168.100.189
	IO_MODULE_DIN_8 (30-1-2)	330000132	1.4.8	1.4.9-2007	OK	192.168.100.189
	ANALOG_DIN (25-1-2)	461000377	0.0.0	1.2.8-1949A	BOOT	192.168.100.189
	RELAY_DIN_2 (22-1-2)	191000162	1.3.12	1.3.12-2006	OK	192.168.100.189
	ROLLER_SH_DIN (23-1-2)	451002574	1.1.10	1.1.10-1949A	OK	192.168.100.189
	DIGITAL_IN_DIN (20-1-2)	181000775	1.2.12	1.2.12-2006	OK	192.168.100.189
	RELAY_DIN_4 (21-1-2)	201000275	0.0.0	1.3.12-2006	DETACHED	192.168.100.189

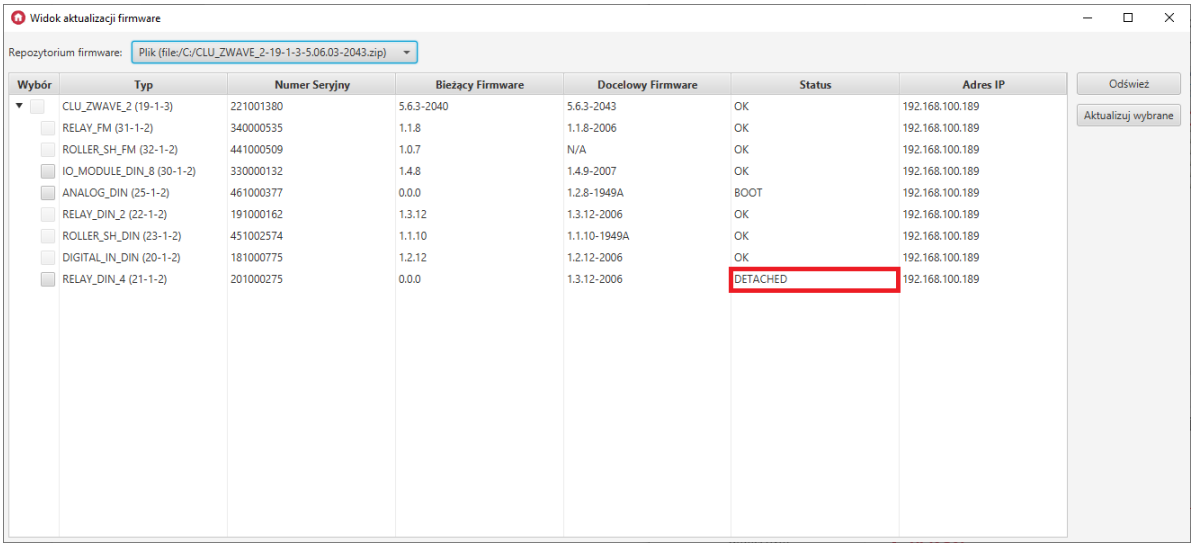
Status: BOOT - Ten status wskazuje, że dany moduł aktualnie jest w bootloaderze. Status taki pojawia się, gdy aktualizacja danego modułu zostanie przerwana. W takiej sytuacji należy ponowić aktualizację danego modułu.

> **Uwaga!** Nie należy wykonywać CLU Discovery, jeżeli moduł posiada status BOOT! Uniemożliwi to wykrycie modułu w procesie Discovery. Jeśli do takiej sytuacji dojdzie, konieczne będzie wymuszenie aktualizacji dla danego modułu patrz pkt VI.10.5..



Wybór	Typ	Numer Seryjny	Bieżący Firmware	Docelowy Firmware	Status	Adres IP
	CLU_ZWAVE_2 (19-1-3)	221001380	5.6.3-2040	5.6.3-2043	OK	192.168.100.189
	RELAY_FM (31-1-2)	340000535	1.1.8	1.1.8-2006	OK	192.168.100.189
	ROLLER_SH_FM (32-1-2)	441000509	1.0.7	N/A	OK	192.168.100.189
	IO_MODULE_DIN_8 (30-1-2)	330000132	1.4.8	1.4.9-2007	OK	192.168.100.189
	ANALOG_DIN (25-1-2)	461000377	0.0.0	1.2.8-1949A	BOOT	192.168.100.189
	RELAY_DIN_2 (22-1-2)	191000162	1.3.12	1.3.12-2006	OK	192.168.100.189
	ROLLER_SH_DIN (23-1-2)	451002574	1.1.10	1.1.10-1949A	OK	192.168.100.189
	DIGITAL_IN_DIN (20-1-2)	181000775	1.2.12	1.2.12-2006	OK	192.168.100.189
	RELAY_DIN_4 (21-1-2)	201000275	0.0.0	1.3.12-2006	DETACHED	192.168.100.189

Status: DETACHED - CLU nie może nawiązać połączenia z danym modułem. W takiej sytuacji należy sprawdzić połączenia magistrali TF-Bus i wykonać restart napięciowy całego systemu.

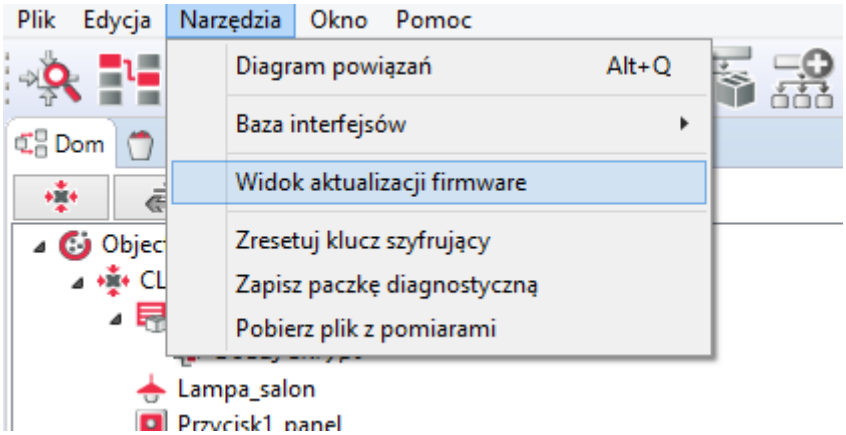


10.5 Procedura wymuszania aktualizacji modułu

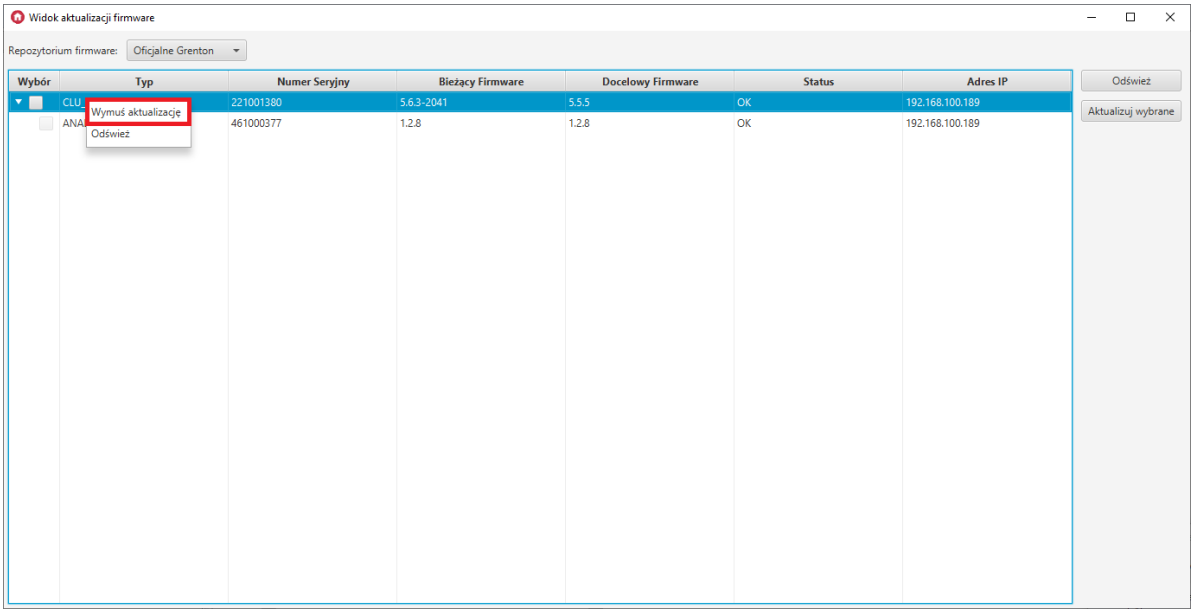
Jeśli prawidłowo podłączony moduł nie jest wykrywalny w procesie Discovery, istnieje możliwość, że firmware dla tego modułu nie zostało prawidłowo wgrane. W takiej sytuacji konieczne jest wymuszenie aktualizacji modułu.

Aby przeprowadzić procedurę wymuszenia aktualizacji modułu, należy:

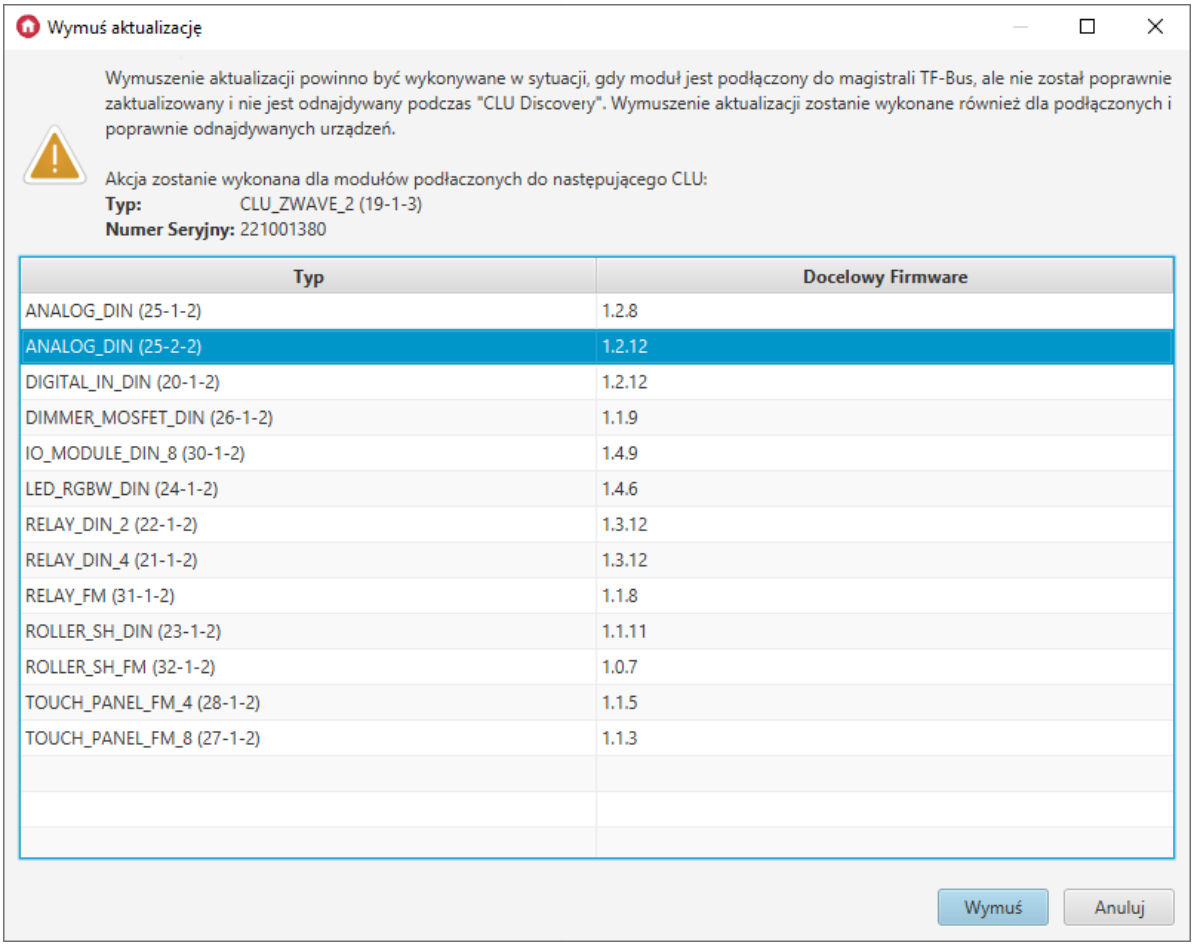
- Z paska menu wybrać `Narzędzia`.
- Wybrać opcję *Widok aktualizacji firmware*:



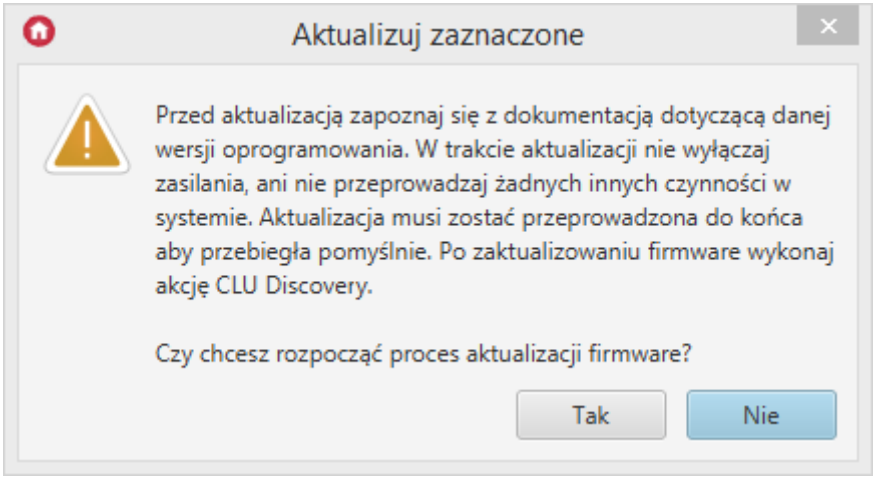
- Kliknąć prawym przyciskiem myszy na moduł CLU i wybrać opcję "Wymuś aktualizację":



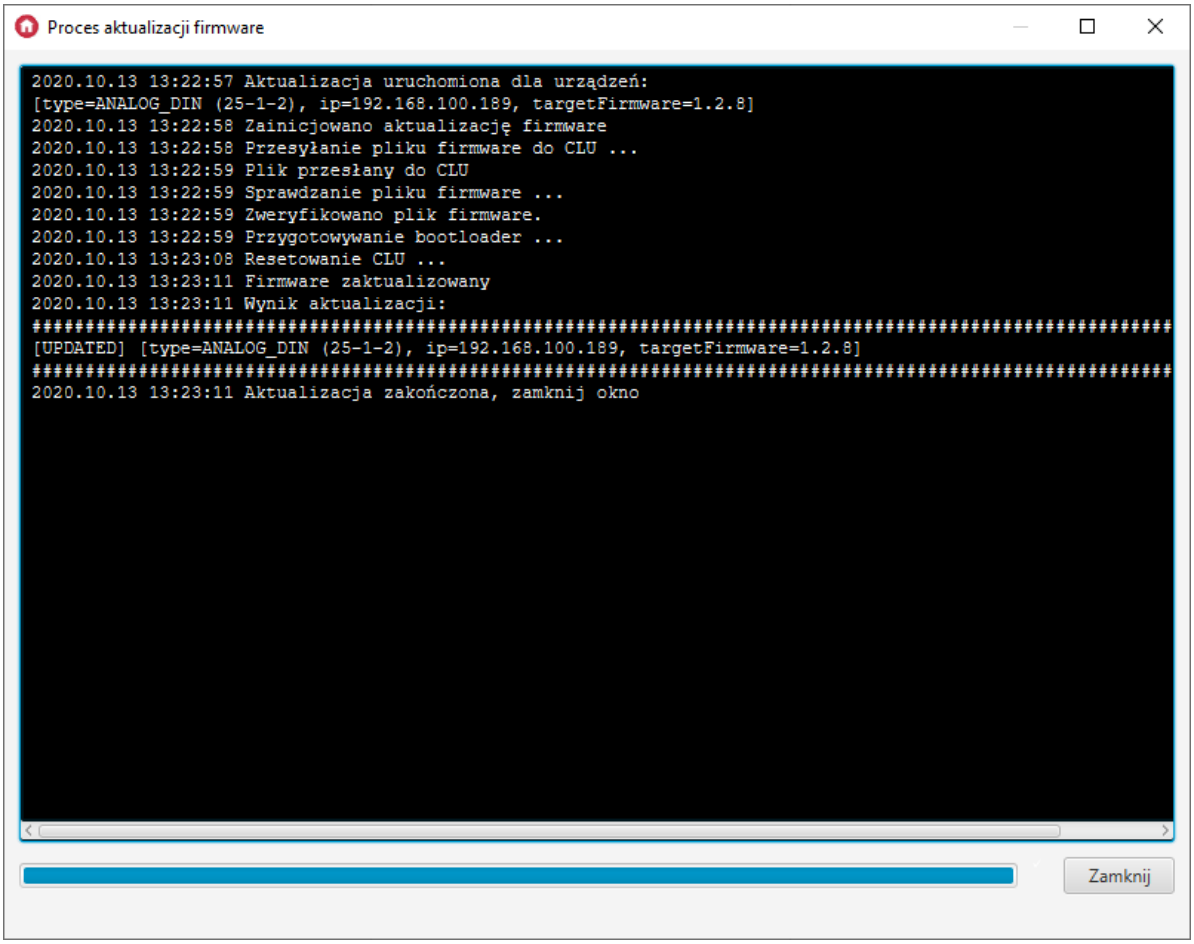
- Wybrać jeden moduł, dla którego ma zostać przeprowadzone wymuszenie aktualizacji, i nacisnąć przycisk `Wymuś`:



- Po zapoznaniu się z komunikatem, wybrać „Tak” aby kontynuować:



- W przypadku poprawnej aktualizacji, przy zaktualizowanym module pojawi się informacja [UPDATED]:



- Po zakończeniu procesu wymuszenia aktualizacji należy wykonać CLU Discovery.

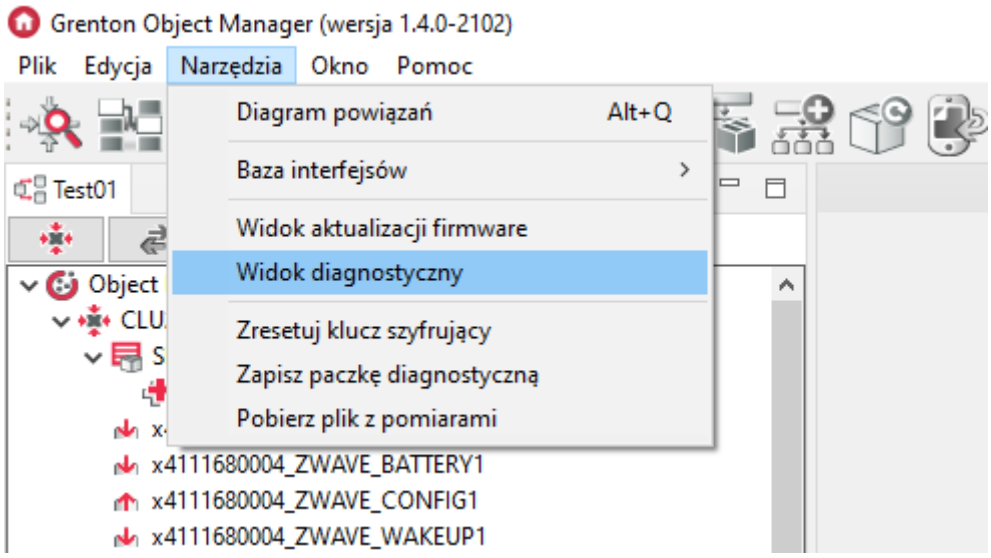
11. Widok diagnostyczny

> **Uwaga!** Aby korzystać z funkcjonalności Widoku diagnostycznego, należy posiadać Object Manager v1.4.0 (lub wyższy) oraz CLUZ posiadające firmware v5.7.1 (lub wyższy).

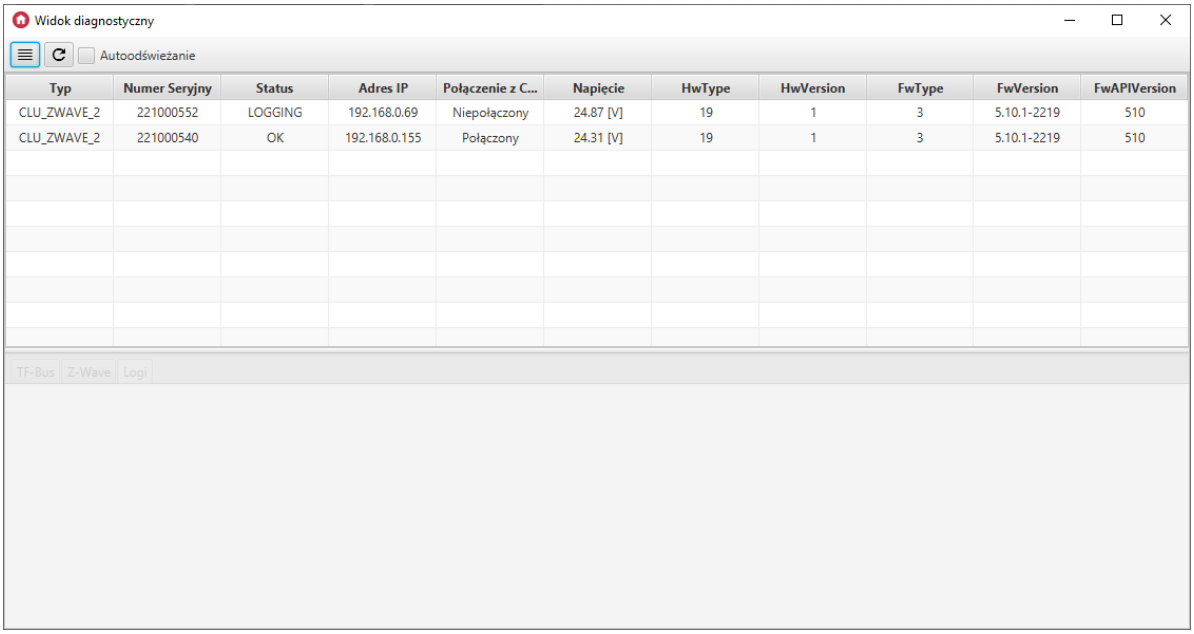
W widoku diagnostycznym prezentowane są informacje o wszystkich CLU w projekcie oraz o podłączonych do nich modułach.

W celu otwarcia Widoku diagnostycznego w Object Managerze należy:

- Z paska menu wybrać `Narzędzia`.
- Wybrać *Widok diagnostyczny*:



Po otwarciu Widoku diagnostycznego ukaże się okno przedstawiające listę wszystkich CLU w projekcie:



W widoku prezentowane są następujące informacje dla CLU:

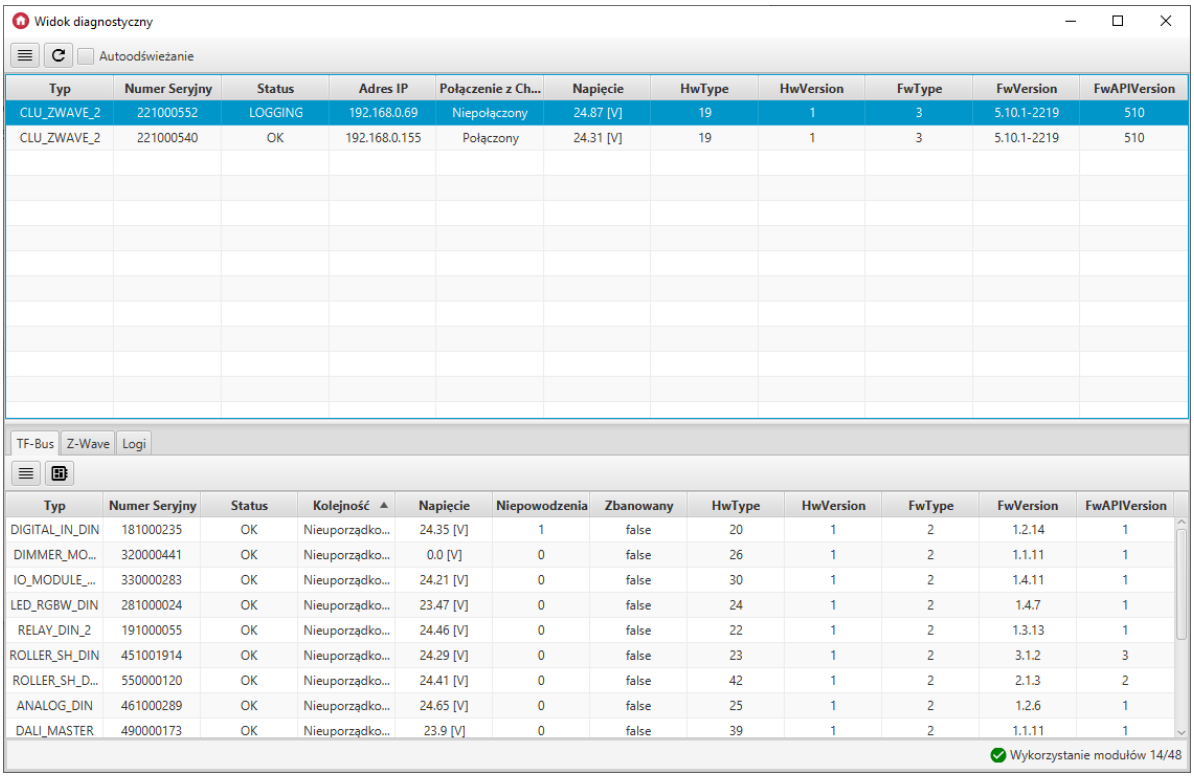
- `Typ` - nazwa typu urządzenia,
- `Numer seryjny` - numer seryjny urządzenia,
- `Status` - status CLU,
- `Adres IP` - adres IP urządzenia,
- `Połączenie z Chmurą` - status połączenia z chmurą,
- `Napięcie` - wartość napięcia zasilania CLU,
- `HwType` - typ hardware,
- `HwVersion` - wersja hardware,
- `FwType` - typ firmware,
- `FwVersion` - wersja firmware,
- `FwAPIVersion` - wersja API firmware.

Statusy CLU:

- `OK` - CLU zwraca dane diagnostyczne,
- `DISCONNECTED` - CLU nie odpowiada,

- `LOGGING` - CLU w trybie logowania,
- `DIAGNOSTICS_OFF` - CLU odpowiada, ale nie wspiera widoku diagnostycznego.

Po kliknięciu na wybrane CLU pojawia się okno ze wszystkimi podłączonymi do niego modułami TF-Bus:



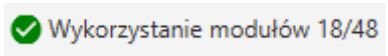
Typ	Numer Seryjny	Status	Adres IP	Połączenie z Ch...	Napięcie	HwType	HwVersion	FwType	FwVersion	FwAPIVersion
CLU_ZWAVE_2	221000552	LOGGING	192.168.0.69	Niepołączony	24.87 [V]	19	1	3	5.10.1-2219	510
CLU_ZWAVE_2	221000540	OK	192.168.0.155	Połączony	24.31 [V]	19	1	3	5.10.1-2219	510

Typ	Numer Seryjny	Status	Kolejność	Napięcie	Niepowodzenia	Zbanowany	HwType	HwVersion	FwType	FwVersion	FwAPIVersion
DIGITAL_IN_DIN	181000235	OK	Nieuporządko...	24.35 [V]	1	false	20	1	2	1.2.14	1
DIMMER_MO...	320000441	OK	Nieuporządko...	0.0 [V]	0	false	26	1	2	1.1.11	1
IO_MODULE_...	330000283	OK	Nieuporządko...	24.21 [V]	0	false	30	1	2	1.4.11	1
LED_RGBW_DIN	281000024	OK	Nieuporządko...	23.47 [V]	0	false	24	1	2	1.4.7	1
RELAY_DIN_2	191000055	OK	Nieuporządko...	24.46 [V]	0	false	22	1	2	1.3.13	1
ROLLER_SH_DIN	451001914	OK	Nieuporządko...	24.29 [V]	0	false	23	1	2	3.1.2	3
ROLLER_SH_D...	550000120	OK	Nieuporządko...	24.41 [V]	0	false	42	1	2	2.1.3	2
ANALOG_DIN	461000289	OK	Nieuporządko...	24.65 [V]	0	false	25	1	2	1.2.6	1
DALI_MASTER	490000173	OK	Nieuporządko...	23.9 [V]	0	false	39	1	2	1.1.11	1

W widoku prezentowane są następujące informacje dla modułów TF-Bus:

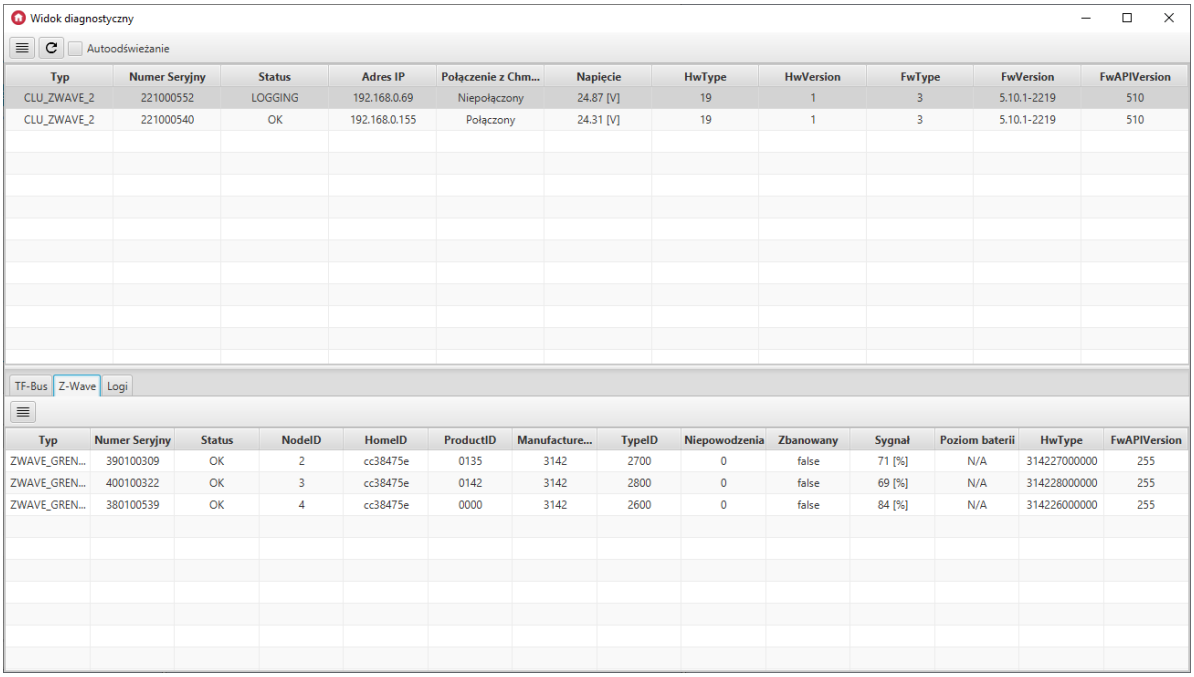
- `Typ` - nazwa typu modułu,
- `Numer Seryjny` - numer seryjny modułu,
- `Status` - stan połączenia z magistralą TF-Bus,
- `Kolejność` - kolejność podpięcia do magistrali TF-Bus (ustawiana manualnie),
- `Napięcie` - wartość napięcia na magistrali dla modułu,
- `Niepowodzenia` - ilość nieudanych odpowiedzi modułu,
- `Zbanowany` - informacja o tym, czy moduł jest zbanowany,
- `HwType` - typ hardware,
- `HwVersion` - wersja hardware,
- `FwType` - typ firmware,
- `FwVersion` - wersja firmware,
- `FwAPIVersion` - wersja API firmware.

W prawym dolnym rogu znajduje się informacja o ilości używanych modułów TF-Bus:



> **Uwaga!** Jeśli dany moduł nie posiada pomiaru napięcia, w kolumnie `Napięcie` zostanie wyświetlona wartość "0.0 [V]".

Jeśli CLU posiada podłączone moduły Z-Wave, będzie możliwe wyświetlenie ich w zakładce `Z-Wave`:



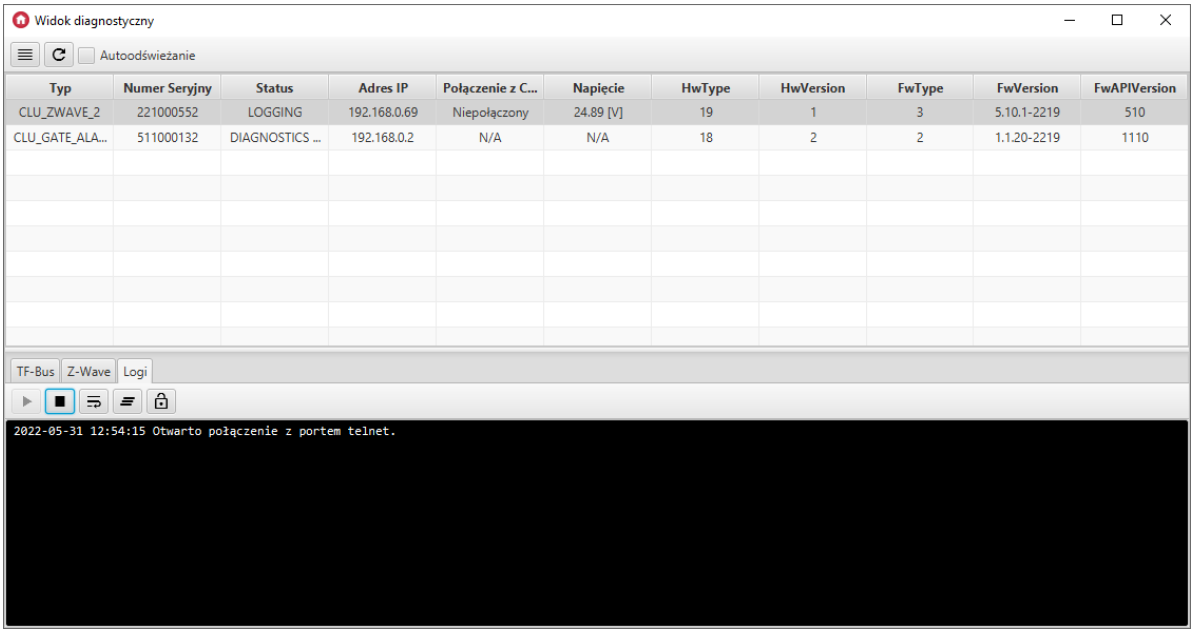
W widoku prezentowane są następujące informacje dla modułów Z-Wave:

- `Typ` - nazwa typu modułu,
- `Numer Seryjny` - numer seryjny modułu,
- `Status` - stan połączenia z CLU,
- `NodeID` - Node ID modułu,
- `HomeID` - Home ID modułu,
- `ProductID` - Product ID modułu,
- `ManufacturerID` - Manufacturer ID modułu,
- `TypeID` - Type ID modułu,
- `Niepowodzenia` - ilość nieudanych odpowiedzi modułu,
- `Zbanowany` - informacja o tym, czy moduł jest zbanowany,
- `Sygnał` - siła sygnału w %,
- `Poziom baterii` - poziom baterii w %,
- `HwType` - typ hardware,
- `FwAPIVersion` - wersja API firmware.

Uwaga! Wartości dla `Sygnał` i `Poziom baterii` aktualizowane są po wybudzeniu modułu. Jeśli moduł jest zbanowany (`Zbanowany = true`) aktualne wartości nie zostaną wyświetlone.

Uwaga! Jeśli moduł Z-Wave nie jest modułem bateryjnym, w kolumnie `Poziom baterii` wyświetlane jest "N/A".

Dla CLU posiadających cechę wbudowaną `TelnetLogLevel` dostępna zakładka `Logi` umożliwiająca rejestrowanie i wyświetlanie logów danego CLU:



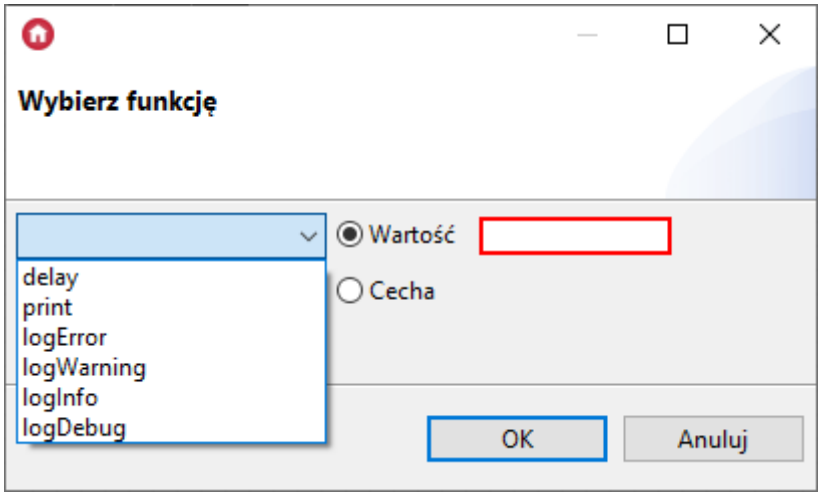
> **Uwaga!** Aby w pełni korzystać z opisanej funkcjonalności logowania, należy posiadać Object Manager w wersji 1.7.0 lub wyższej oraz CLU z firmware 5.10.01 lub wyższym.

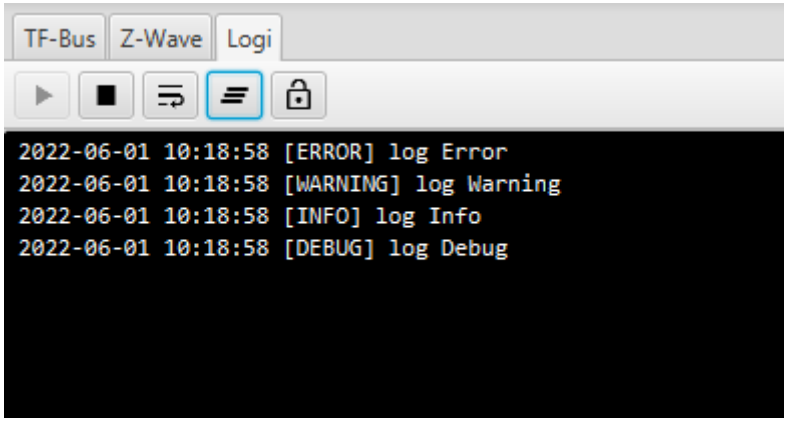
W widoku dostępne jest pole tekstowe prezentujące zarejestrowane logi. Poziom logowania określany za pomocą cechy wbudowanej CLU `TelnetLogLevel` lub metody `SetTelnetLogLevel`:

- OFF,
- ERROR,
- WARNING,
- INFO,
- DEBUG.

> **Uwaga!** W ramach jednego CLU obsługiwane tylko jedno połączenie telnet.
>
> Maksymalna ilość znaków dla jednej linii logu wynosi 1000.

Funkcjonalność wysyłania logów jest dostępna przy użyciu bloków funkcyjnych w skrypcie. Umożliwia wyświetlanie logów z oznaczeniami: [ERROR], [WARNING], [INFO], [DEBUG]. W przypadku użycia funkcji `print`, wartość wyświetlana jest w logowaniu na poziome logowania [DEBUG].



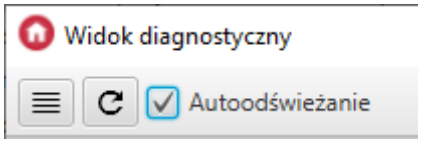


Funkcjonalność logowania dostępna jest również zawsze, gdy CLU znajduje się w trybie Emergency. Możliwe jest wówczas wyświetlenie powodu wejścia w tryb Emergency.

11.1 Konfiguracja widoku diagnostycznego

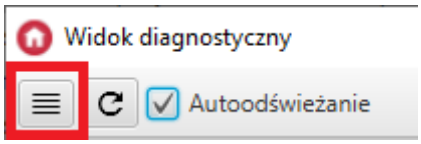
A. Odświeżanie widoku

Widok diagnostyczny odświeża się w momencie naciśnięcia przycisku "Odśwież", lub co 5 sekund, jeśli zaznaczona jest opcja `Autoodświeżanie`.

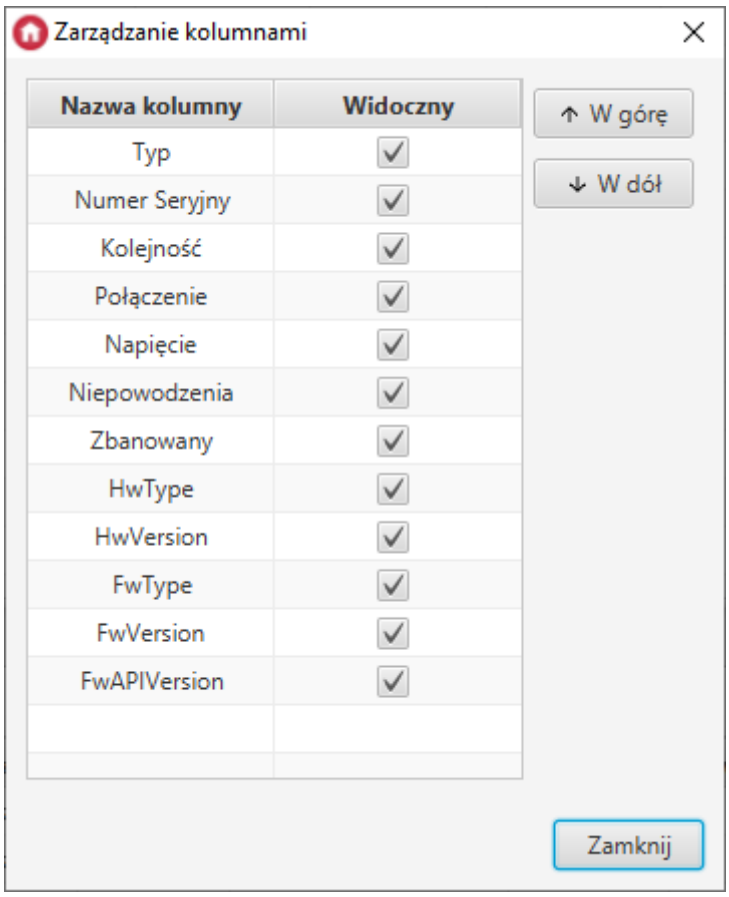


B. Konfiguracja widoku

Widoczność oraz kolejność wyświetlania kolumn można ustawić po naciśnięciu przycisku "Ustawienia kolumn", znajdującego się w lewym górnym rogu okna.



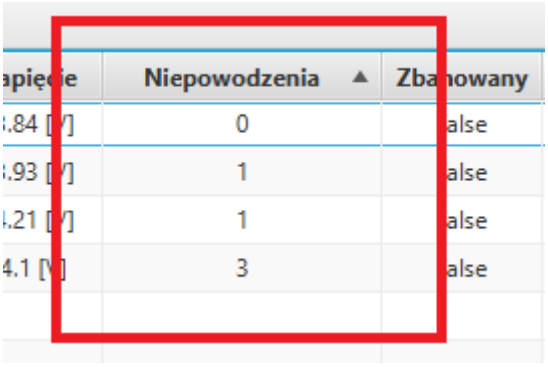
Pojawi się wówczas okno konfiguracji:



Kolejność komun można również zmieniać za pomocą przeciągania ich nazw w oknie głównym.

C. Sortowanie wierszy

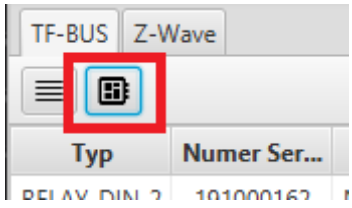
Klikając w nazwę danej kolumny można posortować wiersze w kolejności rosnącej, malejącej lub wrócić do wyświetlania domyślnego.



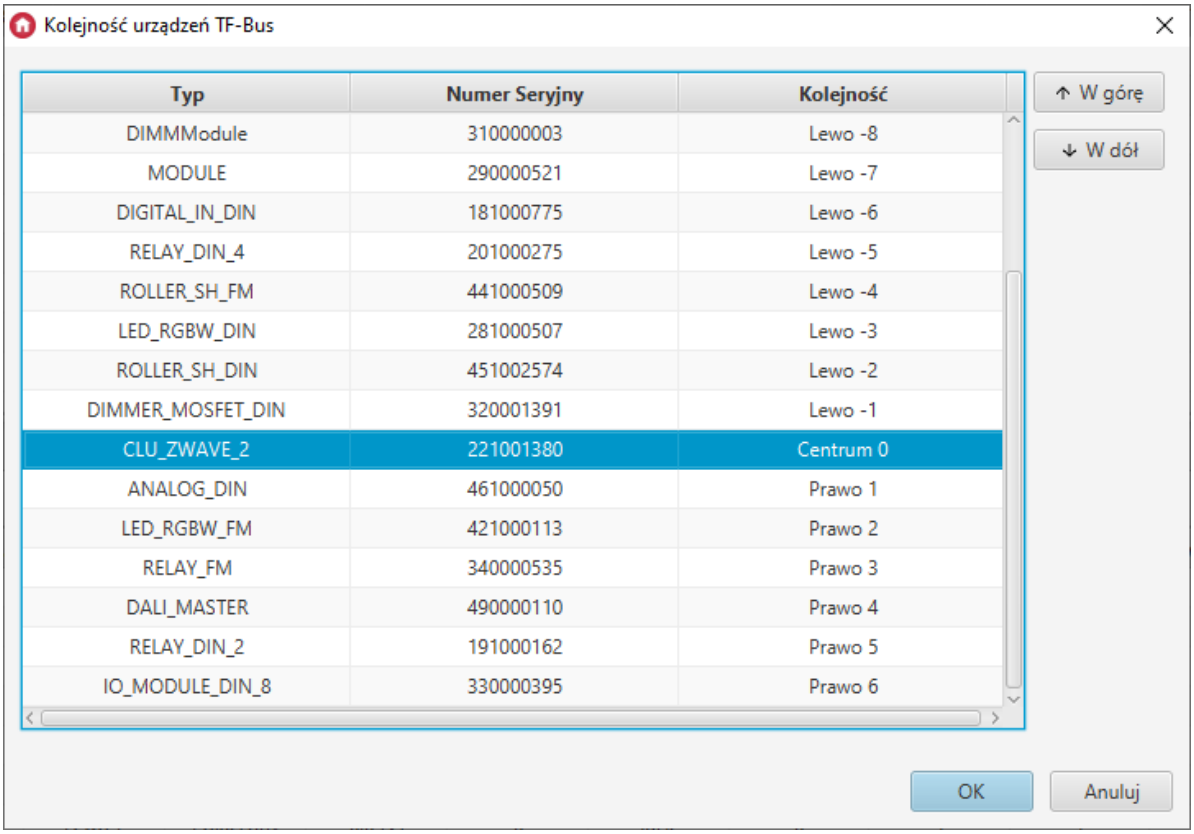
D. Ustawienie kolejności podpięcia do magistrali TF-Bus

Kolejność podpięcia modułów do magistrali TF-Bus należy ustawić ręcznie, inaczej moduły w kolumnie `Kolejność` będą oznaczone jako "Nieuporządkowane".

Aby ustawić kolejność podpięcia modułów do magistrali należy kliknąć przycisk "Kolejność tfbus":

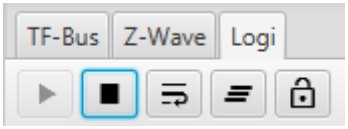


Następnie pojawi się okno, w którym za pomocą przycisków `W górę` i `W dół` można ustawić kolejność modułów względem CLUZ (Centrum 0):



E. Konfiguracja zakładki logowania

W zakładce dostępne przyciski:

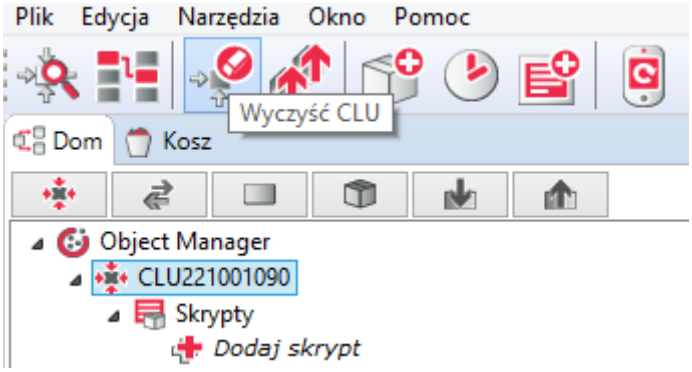


- Rozpocznij logowanie,
- Zatrzymaj logowanie,
- Zawijanie tekstu,
- Wyczyść logi,
- Blokada przewijania.

12. Inne operacje na systemie

Czyszczenie konfiguracji

Użytkownik zawsze ma możliwość wyczyszczenia konfiguracji dowolnego CLU znajdującego się w systemie. W celu wyczyszczenia konfiguracji na wybranym CLU, najpierw musimy je zaznaczyć, a następnie kliknąć w ikonę czyszczenia.



Czyszczenie konfiguracji kasuje wszystkie dokonane zmiany i ustawienia oraz ustawia wartości domyślne (default value).

> **Uwaga!** Po wyczyszczeniu konfiguracji na danym CLU, powiązania pomiędzy obiektami innych CLU, a obiektami CLU czyszczonego, zostaną utracone!

Pobieranie konfiguracji z istniejącego obiektu

Object Manager pozwala na pobranie konfiguracji znajdującej się w istniejącym i działającym systemie. Konfigurację można pobrać tylko i wyłącznie podczas tworzenia nowego czystego projektu - nie jest możliwe pobranie konfiguracji dla projektu, który posiada już jakieś dane.

Dodawanie nowego modułu CLU lub IOM

Po zainstalowaniu nowego modułu należy dodać go do systemu. Moduł musi zostać wpięty do magistrali systemowej (przed wpięciem nowego modułu należy odłączyć zasilanie magistrali). W przypadku modułów Z-Wave należy dodać je do kontrolera - patrz pkt VI.6.1.. Po poprawnym zainstalowaniu modułu, należy uruchomić CLU Discovery, spowoduje to automatyczne wyszukanie i dodanie nowego modułu. Jeśli w systemie istnieją nieużywane wejścia/wyjścia, system uruchomi listę umożliwiającą przypisanie nieaktywnych wejść/wyjść do wejść/wyjść z nowego modułu. Po wykonaniu powyższej procedury moduł pojawi się na liście obiektów.

Wymiana modułu IOM (wejść/wyjść)

Jeśli dany moduł zostanie wymieniony na inny, ale o takich samych parametrach (taki sam typ i taka sama ilość wejść/wyjść), należy dokonać również wymiany modułu w projekcie w programie Object Manager. Po poprawnym zainstalowaniu i podpięciu modułu należy uruchomić funkcję CLU Discovery. System automatycznie wyszuka i rozpozna nowy moduł, oraz automatycznie przypisze do niego wejścia/wyjścia ze „starego" modułu. Po wyszukaniu zostanie wyświetlona lista z przypisaniem wejść/wyjść pomiędzy wymienionymi modułami oraz opcja potwierdzenia i zaakceptowania zmiany. W przypadku zaakceptowania zmian, na liście obiektów nic się nie zmieni, a wszystkie przypisania dokonają się automatycznie. Brak akceptacji spowoduje pojawienie się na liście obiektów nowych pozycji, przy czym jednocześnie również wyświetlane będą nieaktywne wejścia/wyjścia (oznaczone kolorem szarym).

Przepięcie modułu z jednego CLU do innego znajdującego się w tym samym systemie.

W sytuacji, gdy zachodzi konieczność przełączenia modułu IOM z jednego CLU do drugiego, należy przepiąć fizycznie moduł (przełączyć kable), a następnie wykonać funkcję CLU Discovery, która uaktualni listę modułów we wszystkich CLU.

VII. Zaawansowane funkcje konfiguracyjne

1. Kontenery

W celu łatwiejszego zarządzania dostępnymi wejściami/wyjściami OM posiada funkcjonalność kontenerów, która pozwala grupować wejścia/wyjścia zgodnie z potrzebami użytkownika.

Kontener może być wykorzystany na przykład do posortowania wejść/wyjść zgodnie z ich funkcjami (oświetlenie, ogrzewanie itp.) lub też umiejscowieniem w budynku (salon, kuchnia itp.).

Nowy kontener tworzy się poprzez: kliknięcie w menu górnym ikony kontenera lub kliknięcie prawym przyciskiem myszy na drzewku obiektów i wybranie opcji *Nowy kontener*. W drzewie, na poziomie kontenera głównego, pojawi się ikona nowego kontenera. Należy nadać mu nazwę, ale trzeba pamiętać, że nie może zawierać polskich liter.

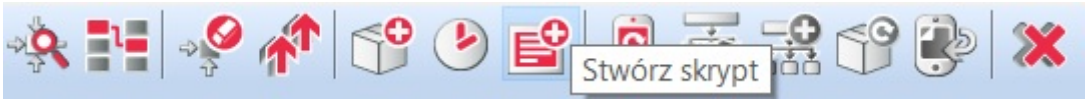
Wejścia/wyjścia do kontenerów przypisuje się przez: przeciągnięcie z CLU lub po kliknięciu na nim prawym przyciskiem myszy i wybraniu opcji *Przenieś do kontenera*.

2. Skrypty

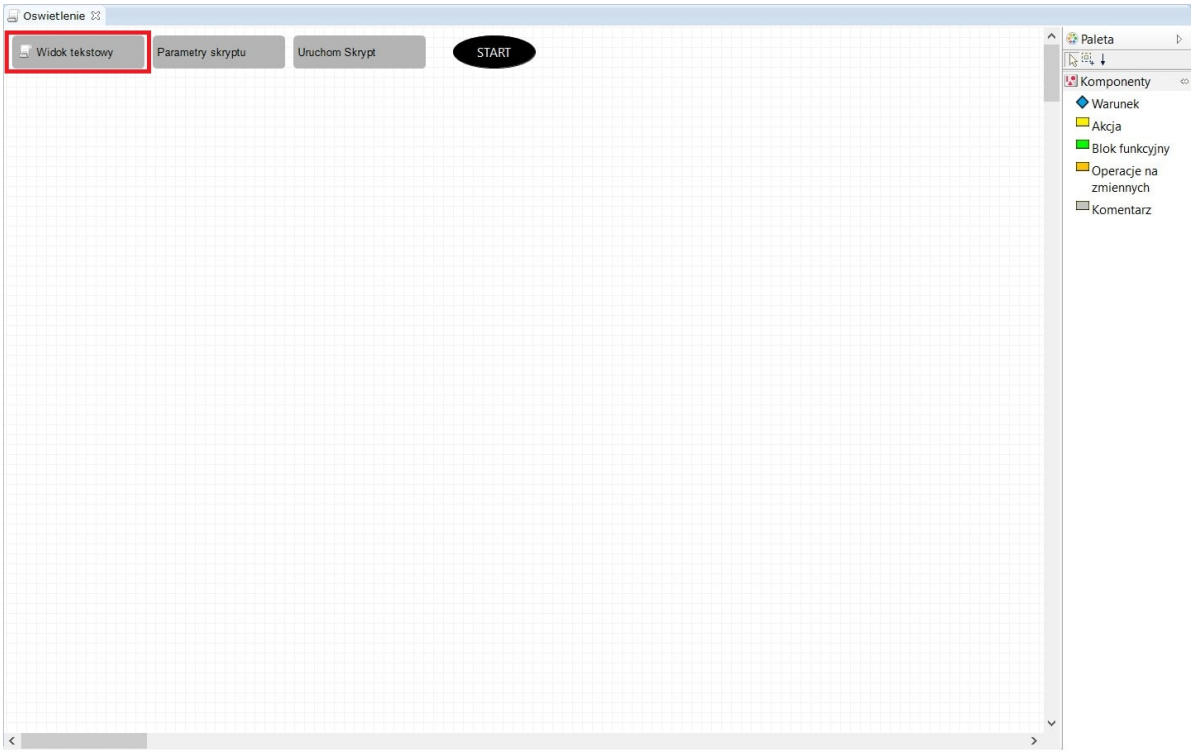
Skrypty umożliwiają tworzenie bardzo rozbudowanej logiki z wykorzystaniem instrukcji warunkowych, pętli czy zmiennych. Dają tym samym możliwość tworzenia zaawansowanych scen, które modyfikują działanie w zależności od warunków zewnętrznych.

Utworzone skrypty widoczne są w systemie jako metody CLU i mogą być wywoływane poprzez przypisanie do zdarzeń dowolnego obiektu, a także wywoływane z poziomu innych skryptów.

Skrypty tworzy się klikając na CLU, na którym skrypt ma być przechowywany. W tym celu należy uruchomić, a następnie wybrać opcję `Stwórz skrypt` w menu akcji, zgodnie z poniższym rysunkiem:



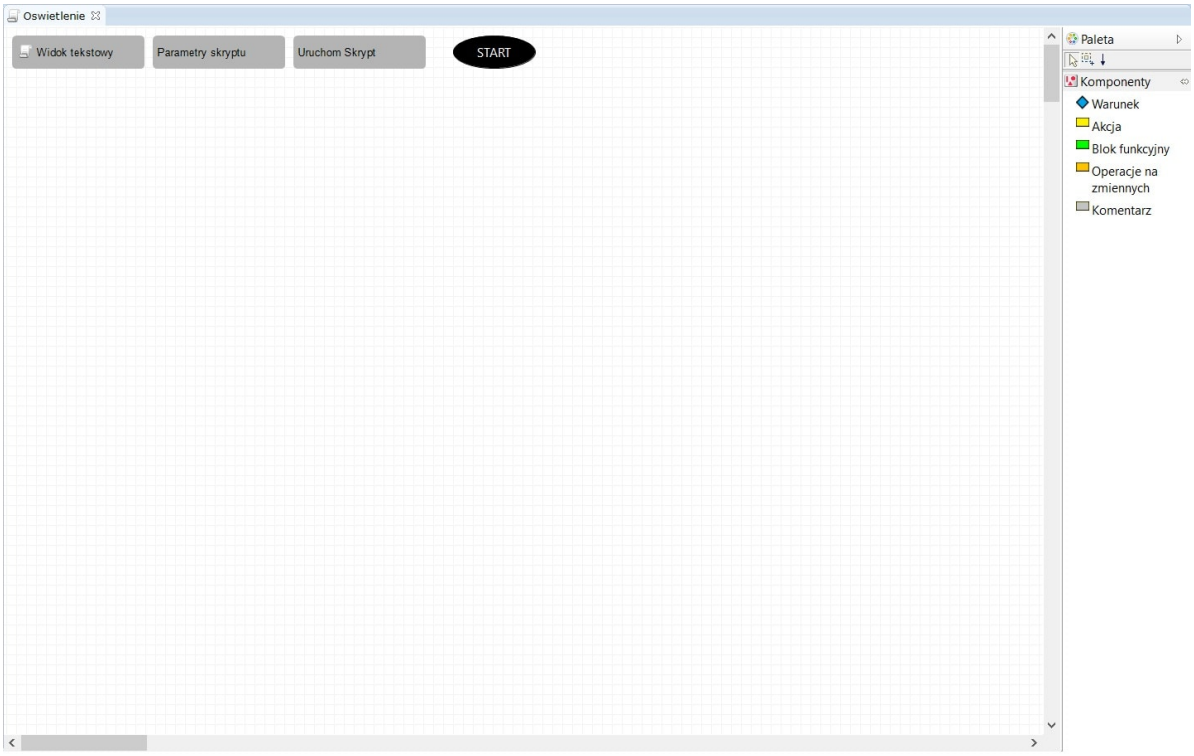
Po podaniu nazwy skryptu (bez polskich znaków) w zakładce otworzy się Script Builder, przy pomocy którego można stworzyć skrypt. Script Builder może pracować w dwóch trybach: graficznym i tekstowym. Po utworzeniu nowego skryptu Script Builder uruchamia się w trybie graficznym, z którego możemy przejść do trybu tekstowego klikając przycisk `Widok tekstowy`, zgodnie z poniższym rysunkiem.



Uwaga! Przejście z trybu graficznego w tryb tekstowy jest nieodwracalne. Gdy w edytorze graficznym stworzony został jakiś skrypt, zostanie on przekonwertowany na postać tekstową, jednak po dokonaniu edycji w edytorze tekstowym powrót do edytora graficznego nie będzie możliwy.

2.1 Tworzenie skryptów w trybie graficznym

Po otworzeniu pojawia się czysty arkusz.



Z prawej strony arkusza znajduje się lista komponentów. Rozkazy można wstawiać z listy poprzez przeciągnięcie do arkusza. Po upuszczeniu rozkazu na arkuszu otwiera się okno dialogowe, które umożliwia podanie parametrów rozkazu lub instrukcji warunkowych. Po dodaniu do arkuszu nowego komponentu, automatycznie tworzone jest połączenie pomiędzy ostatnim wstawionym komponentem (lub polem `Start`, jeśli jest to pierwszy komponent), a obecnie wstawianym. Rozkazy wykonywane są w kolejności zgodnej z połączeniami - rozpoczynając od startu. Zmiana kolejności wykonywania rozkazów jest możliwa po usunięciu istniejącego połączenia i poprowadzeniu nowego (zgodnie z żądaną kolejnością), za pomocą narzędzia `Połączenie`.



> **Uwaga!** Pozostawienie na arkuszu komponentu, który nie jest połączony z innymi komponentami, traktowane jest jako błąd i zostanie wyświetlone jako błąd konfiguracji dla CLU, na którym skrypt został utworzony.

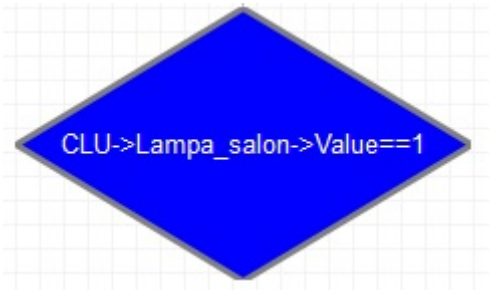
W Script Builderze wykorzystywane są następujące komponenty:

A. Akcja



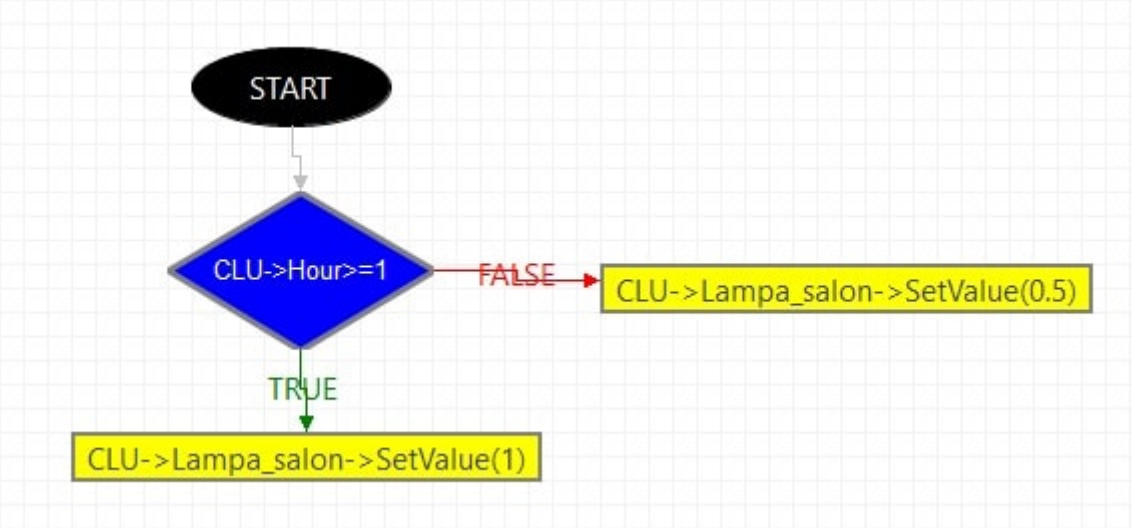
Blok, w którym wpisywany jest rozkaz, jaki ma zostać wykonany. Rozkazem może być wywoływanie metody, ale również zmiana wartości oraz wywołanie skryptu. Po przeciągnięciu ikony akcji do arkusza otwiera się okno z listą obiektów oraz ich metodami. Skrypty są dostępne na liście po kliknięciu na CLU, na którym są umieszczone jako metody CLU.

B. Warunek

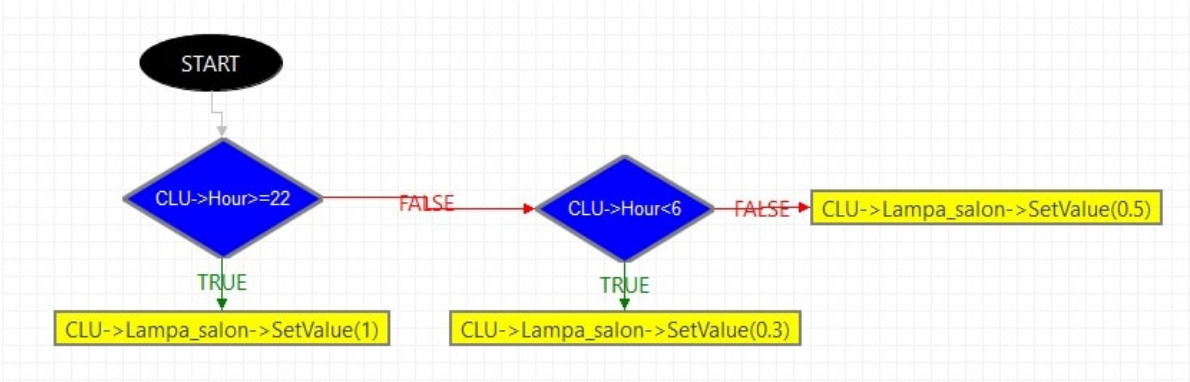


Blok logiczny realizujący funkcję `IF then ELSE`. Użycie tego bloku umożliwia uzależnienie wykonywanej akcji od warunków, np. jeśli jest ciemno, to zapal światło, jeśli nie, to wyłącz. Po przeciągnięciu do arkusza, w parametrach bloku należy wpisać warunek, jaki ma być spełniony. Po dodaniu komponentu *Warunek* należy wstawić komponent *Akcja* lub *Operacje na zmiennych.* OM zapyta, czy polecenie ma być wykonane, gdy warunek zostanie spełniony (`true`) lub odwrotnie - gdy nie zostanie spełniony (`false`). Do warunku można dołączyć dwie akcje, z których jedna zostanie wykonana, gdy warunek będzie spełniony, a w innym przypadku zostanie wykonana druga akcja. Aby zmienić przypisanie `true` / `false` kliknij dwukrotnie na jedną ze strzałek wychodzących z warunku.

Poniższy rysunek pokazuje prostą instrukcje warunkową, która uzależnia natężenie oświetlenia w zależności od godziny.



Warunki mogą być łączone kaskadowo, dzięki czemu istnieje możliwość implementacji operatora `and`, czyli wykonywania akcji, gdy kilka warunków zostanie spełnionych. Poniższy diagram pokazuje przykład zastosowania takiego łączenia:



Warunki mogą porównywać dowolną cechę obiektu lub parametr skryptu z liczbą, tekstem, inną cechą lub parametrem skryptu.

C. Blok funkcyjny

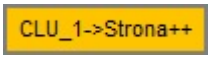


Zawiera instrukcje wywoływane wewnątrz skryptu, które mogą być wykorzystane przy tworzeniu bardziej zaawansowanych scen (funkcja *Delay*) oraz debuggingu (funkcja *Print*). Po przeciągnięciu ikony bloku do arkusza otwiera się okno wyboru. Okno zawiera listę bloków funkcyjnych, na której znajdują się:

- **DELAY** Umożliwia wstawienie zwłoki czasowej pomiędzy kolejnymi instrukcjami w trakcie realizowania skryptu.
- **PRINT** Komenda wywołująca wyświetlenie z góry zadeklarowanego tekstu w debuggingu.
- **logError, logWarning, logInfo, logDebug**

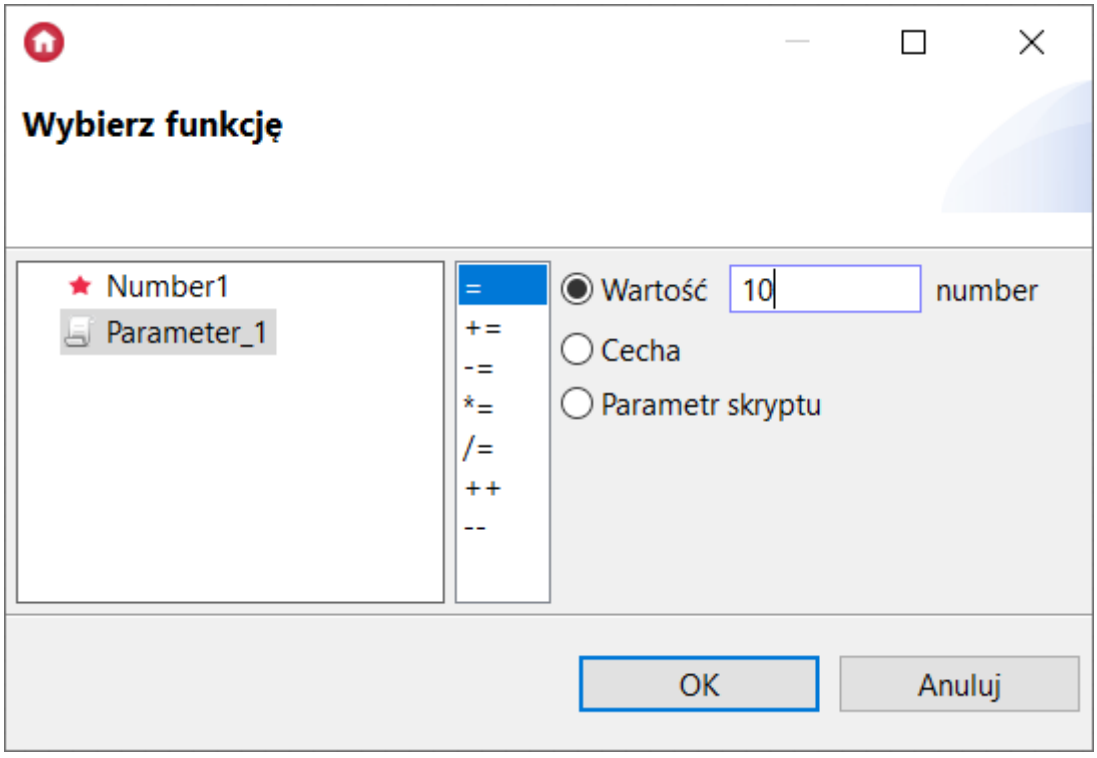
  Komendy umożliwiające wyświetlenie z góry zadeklarowanego tekstu w trybie logowania z oznaczeniami kolejno: [ERROR], [WARNING], [INFO], [DEBUG].

D. Operacja na zmiennych



Blok ten umożliwia tworzenie skomplikowanych funkcji logicznych z wykorzystaniem zmiennych. By wykorzystać zmienne w skrypcie, muszą na początku zostać zdeklarowane. Zmienne można zadeklarować w parametrach skryptu oraz w cechach użytkownika CLU. Zmienna, zdeklarowana jako parametr skryptu, może być wykorzystywana wewnątrz skryptu - na potrzeby dokonywania obliczeń w czasie wykonywania tego skryptu. Dane przechowywane wewnątrz tej zmiennej nie są dostępne na zewnątrz. By dane zawarte w zmiennej mogły być przechowywane lub wykorzystywane poza skryptem, należy skorzystać z cechy użytkownika CLU.

W przypadku cechy użytkownika oraz parametru skryptu typu `number`, dostępne są określone działania matematyczne.



Opis działań na podstawie powyższego przykładu (przy podanej wartości `number` = 10):

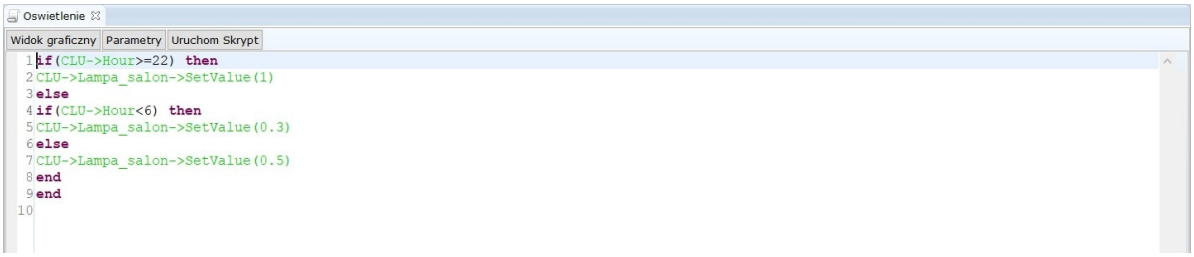
Działanie	Opis
=	Ustawienie liczby 10 jako wartości cechy użytkownika / parametru skryptu.
+=	Dodawanie liczby 10 do wartości cechy użytkownika / parametru skryptu.
-=	Odejmowanie liczby 10 od wartości cechy użytkownika / parametru skryptu.
*=	Mnożenie liczby 10 przez wartość cechy użytkownika / parametru skryptu.
/=	Dzielenie wartości cechy użytkownika / parametru skryptu przez liczbę 10.
++	Powiększenie wartości cechy użytkownika / parametru skryptu o +1.
--	Pomniejszenie wartości cechy użytkownika / parametru skryptu o -1.

2.2. Tworzenie skryptów w trybie tekstowym

Drugą metodą tworzenia skryptów jest użycie edytora tekstowego, który daje praktycznie nieograniczone możliwości tworzenia skryptów z wykorzystaniem instrukcji LUA, rozszerzonych o możliwość używania adresów obiektów interfejsu logicznego.

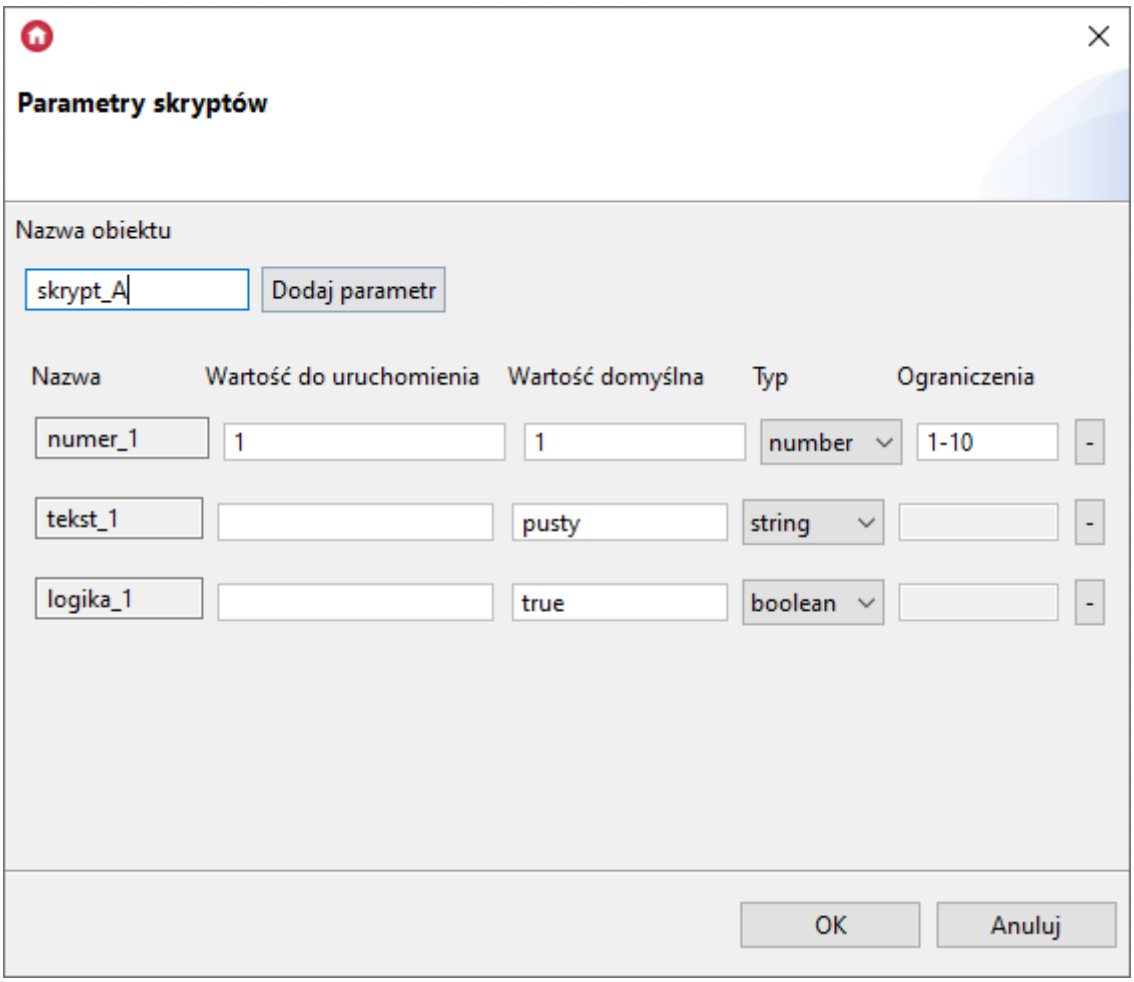
Adresy interfejsu logicznego traktowane są jako funkcje i mogą być wywoływane i używane jako parametry w instrukcjach warunkowych, pętlach itp.

Poniższy skrypt pokazuje sposób użycia adresów interfejsu logicznego w skryptach:



2.3. Parametry skryptu

Skrypty mogą posiadać parametry startowe, które są przekazywane podczas ich wywołania (np. w zdarzeniu), a następnie można z nich korzystać wewnątrz skryptu (np. w instrukcjach warunkowych). Parametry skryptu tworzone są poprzez kliknięcie w opcję `Parametry skryptu`, w następstwie czego otworzy się formatka, w której należy wybrać `Dodaj parametr`, i zdefiniować nazwę, wartość do uruchomienia, wartość domyślną, typ oraz ograniczenia. Aby usunąć zmienną, kliknij przycisk `-` po prawej stronie.



> **Uwaga!** Nazwy zmiennych nie mogą zawierać spacji oraz zaczynać się od cyfry lub znaku.

Wartość do uruchomienia - to wartość, która jest przyjmowana do uruchomienia skryptu przy użyciu przycisku `Uruchom skrypt` w programie Object Manager.

Wartość domyślna - to wartość parametru, która zostanie przyjęta, jeśli podczas wywołania skryptu parametr nie zostanie określony.

Typ - pozwala zdefiniować rodzaj danych, które będą przechowywane w parametrze:

- **string** – dla danych tekstowych;
- **number** – dla danych liczbowych;
- **boolean** – dla zmiennych logicznych `true` / `false`.

Ograniczenia - dla parametrów liczbowych można wprowadzić ograniczenie maksymalnej i minimalnej wartości, jaką może przyjąć zmienna. W przypadku próby wywołania skryptu z parametrem spoza tego zakresu, skrypt zostanie wywołany w wartością domyślną parametru. Ograniczenia podajemy w formacie x-y, gdzie x i y to wartości minimalna i maksymalna ograniczenia. Po wprowadzeniu ograniczenia należy uzupełnić pole *Wartość do uruchomienia*.

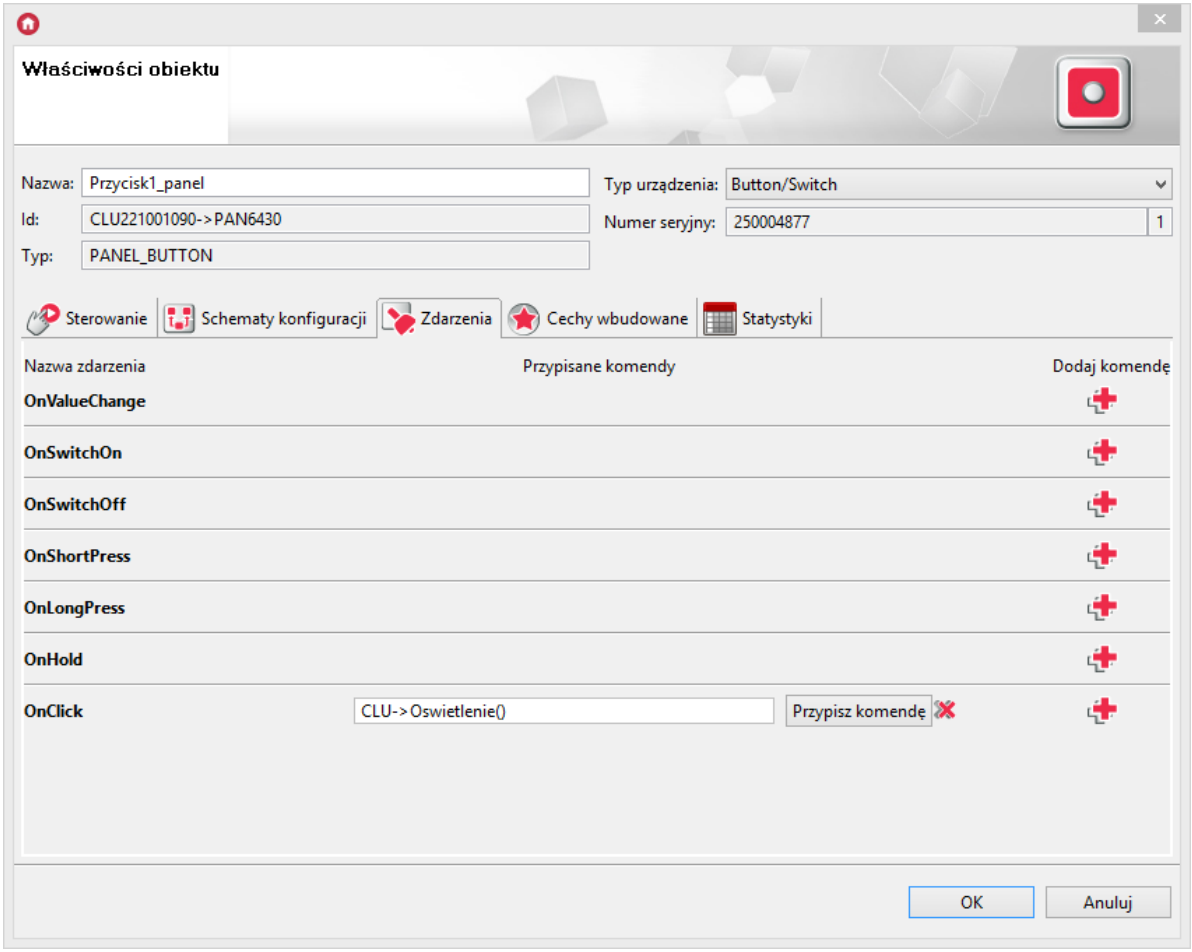
> **Uwaga!** Parametr skryptu przechowuje wartości, które mogą być wykorzystywane tylko wewnątrz niego (lokalnie). Wartości te nie są dostępne w innych skryptach. Jeśli konieczne jest zapisanie wartości lub ciągu znaków do wykorzystania w innych obszarach, należy wykorzystać cechy użytkownika dostępne w CLU lub też przekazać wartość do drugiego skryptu przy pomocy jego parametru.

> **Uwaga!** Zmienne lokalne w jednym skrypcie obsługują wyłącznie CLU, na którym skrypt został stworzony. Aby zmienna obsługiwała inne CLU, należy zdefiniować zmienną użytkownika na CLU, np.:
>
> (CLU_A - CLU, na którym został stworzony skrypt)
>
> ```
> CLU_A->Lampa1->SetValue(zmienna_lokalna)
> CLU_B->Lampa2->SetValue(zmienna_użytkownika)
> ```

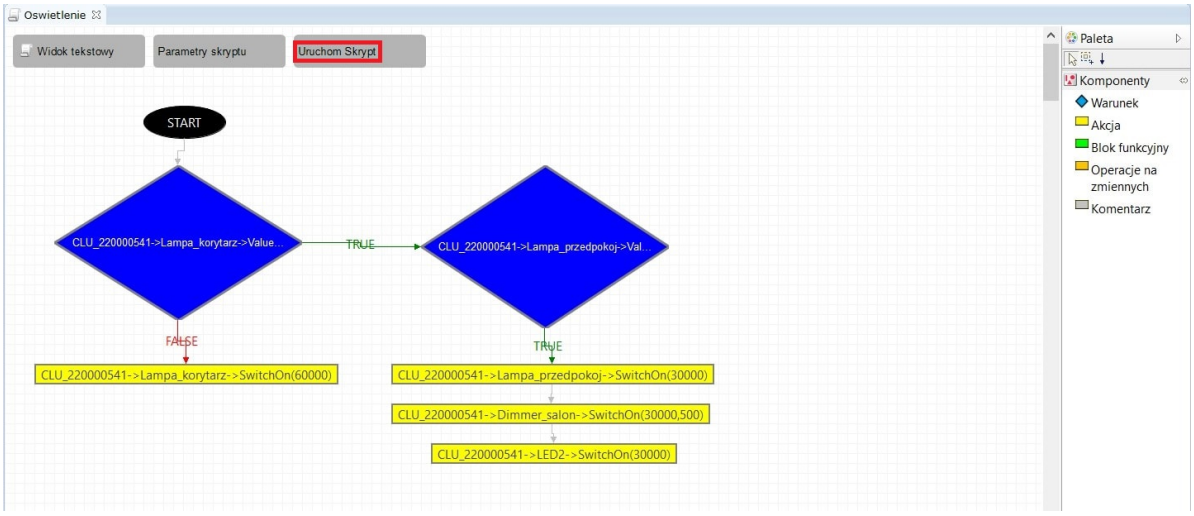
2.4. Wywoływanie skryptów

Skrypty są widoczne i traktowane jako metody CLU. Mogą być wywoływane ze zdarzeń z dowolnego obiektu oraz z bloku akcji w innym skrypcie (na identycznych zasadach jak inne metody).

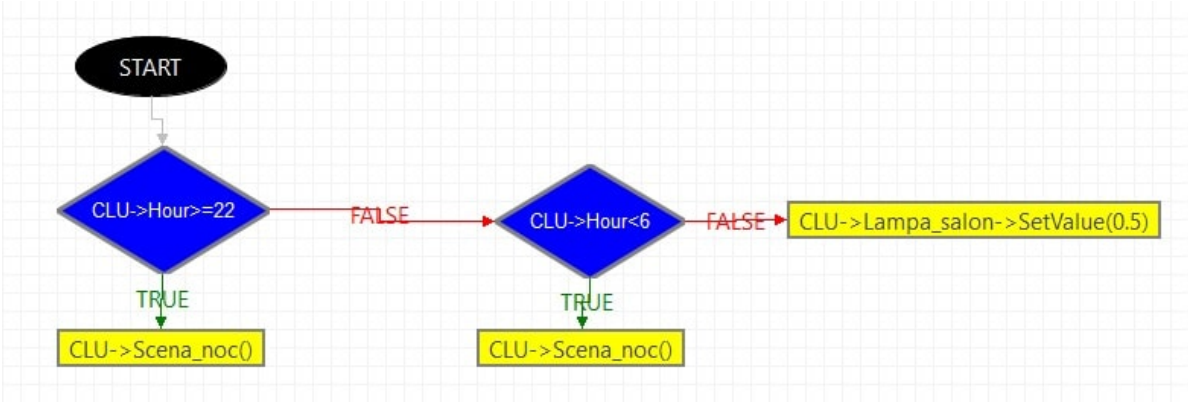
- **Wywołanie poprzez zdarzenie** Poniższy rysunek przedstawia przypisanie skryptu do włącznika, skrypt zostanie uruchomiony po jego wciśnięciu.



- **Wywołanie z poziomu skryptu** Poniższy rysunek ukazuje sposób wywołania z poziomu skryptu za pomocą przycisku `Uruchom skrypt`.



- **Wywołanie z innego skryptu** Poniższy rysunek pokazuje fragment diagramu, w którym wywołano inny skrypt przy pomocy bloku akcji.



- **Wywołanie skryptu z parametrem**

Aby określić parametry wejściowe dla zmiennych podczas wywoływania skryptu, należy wpisać je w nawiasie w prawidłowej kolejności:

```
CLU->skrypt(12,"tekst",true)
```

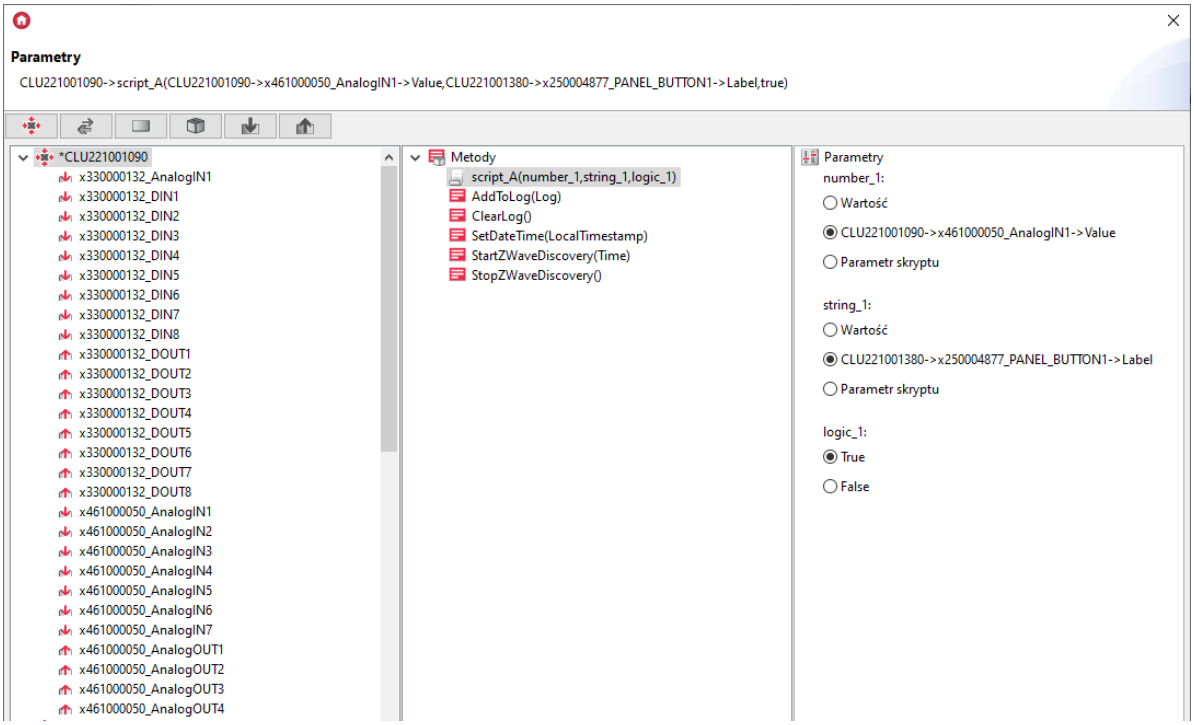
gdzie: określono parametry kolejno dla zmiennej typu number, string oraz boolean.

Aby do zmiennych lokalnych przypisać określone cechy, należy wpisać pełne ścieżki danych cech:

```
CLU->skrypt_A(CLU->AnalogIN1->Value,CLU->BUTTON1->Label,CLU->CloudConnection)
```

gdzie: określono parametry kolejno dla zmiennej typu number, string oraz boolean.

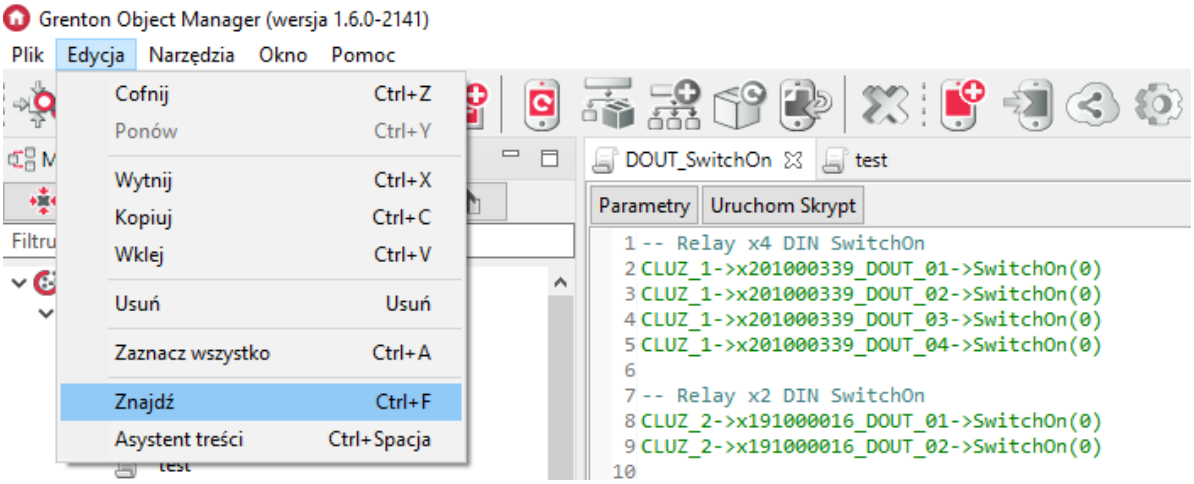
Parametry wejściowe można w prosty sposób określić za pomocą okna Parametrów:



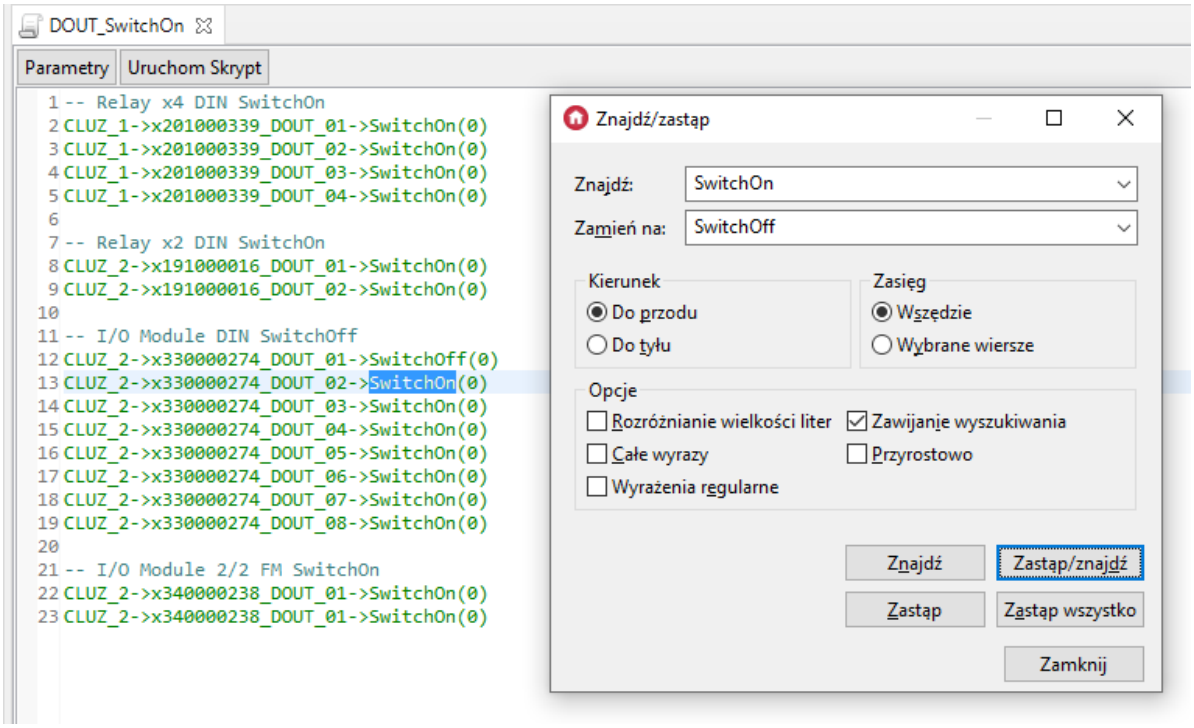
2.4. Funkcja Znajdź / Zastąp

> **Uwaga!** Funkcja *Znajdź / Zastąp* dostępna jest w Object Manager w wersji 1.6.0 lub wyższej.

W przypadku skryptów tekstowych możliwe jest wykonywanie operacja znajdowania oraz zamieniania danych ciągów znaków w skrypcie. Funkcja Znajdź / Zastąp dostępna jest z poziomu Menu `Edycja -> Znajdź / Zastąp` lub można ją wywołać w skrypcie przy pomocy skrótu klawiszowego `Ctrl+F`



W przypadku wyszukiwania danego ciągu znaków, należy podać go w polu tekstowym `Znajdź` natomiast przy wyszukiwaniu i zamianie, należy podać nową frazę również w polu tekstowym `Zamień na`. W oknie dostępne są opcje określające specyfikę wykonywania wyszukiwania danej frazy m.in. określenie kierunku wyszukiwania, obszaru dla którego wykonywane jest wyszukiwanie czy rozróżnianie wielkości liter.



Dostępne akcje:

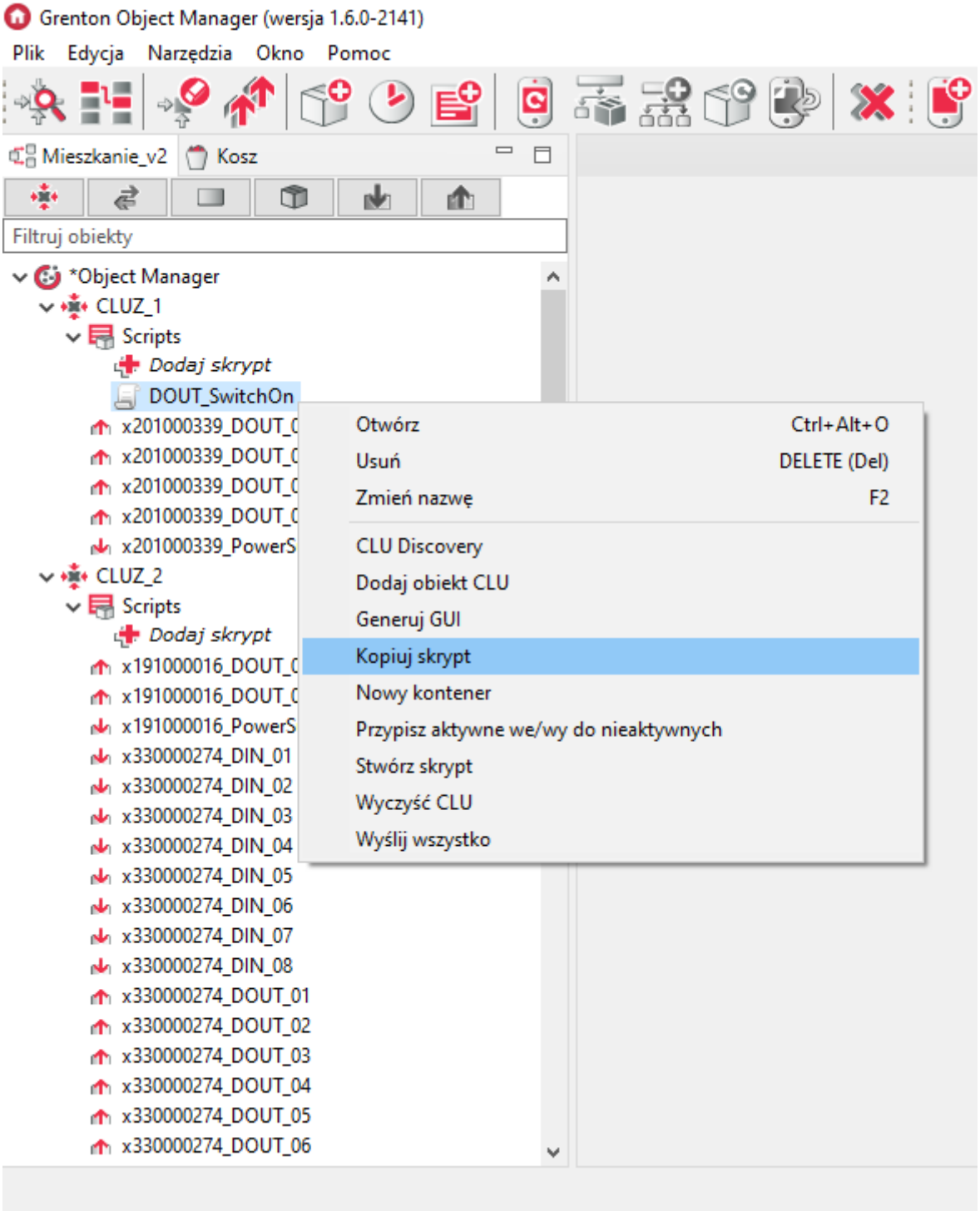
- *Znajdź* - wyszukiwana jest wpisana fraza w polu *Znajdź*;
- *Zastąp* - znaleziona fraza jest zastępowana tekstem wpisanym w polu *Zamień na;*
- *Zastąp/Znajdź* - wykonywana jest zamiana wskazanej frazy a następnie wyszukiwana jest kolejny tekst (wpisany w polu *Znajdź*);
- *Zastąp wszystko* - zamiana następuje w całym skrypcie dla podanej frazy.

> **Uwaga!** Funkcja *Znajdź / Zastąp* nie jest dostępna w przypadku skryptów graficznych.

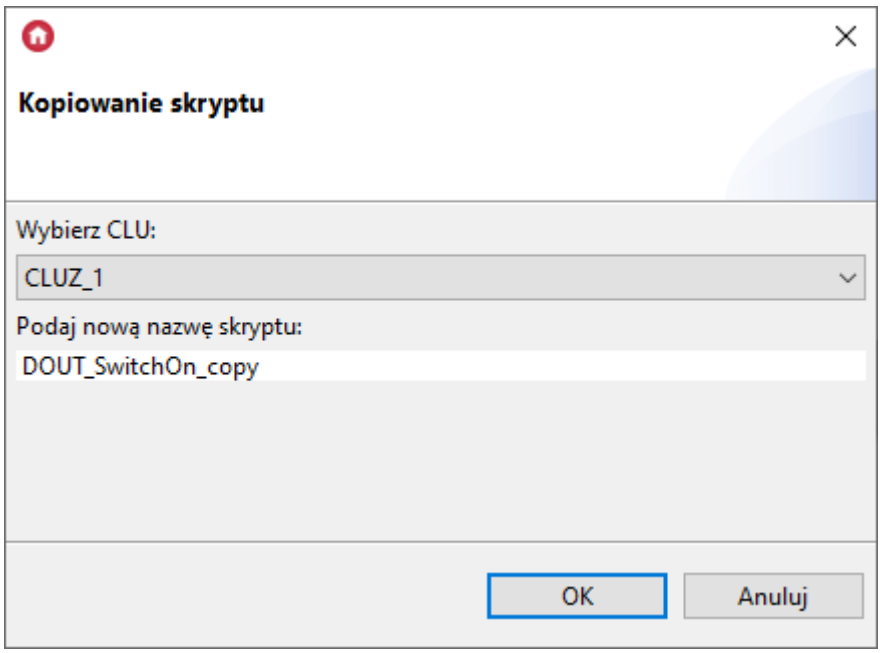
2.5. Kopiowanie skryptów

> **Uwaga!** Opcja kopiowania dostępna jest w Object Manager w wersji 1.6.0 lub wyższej.

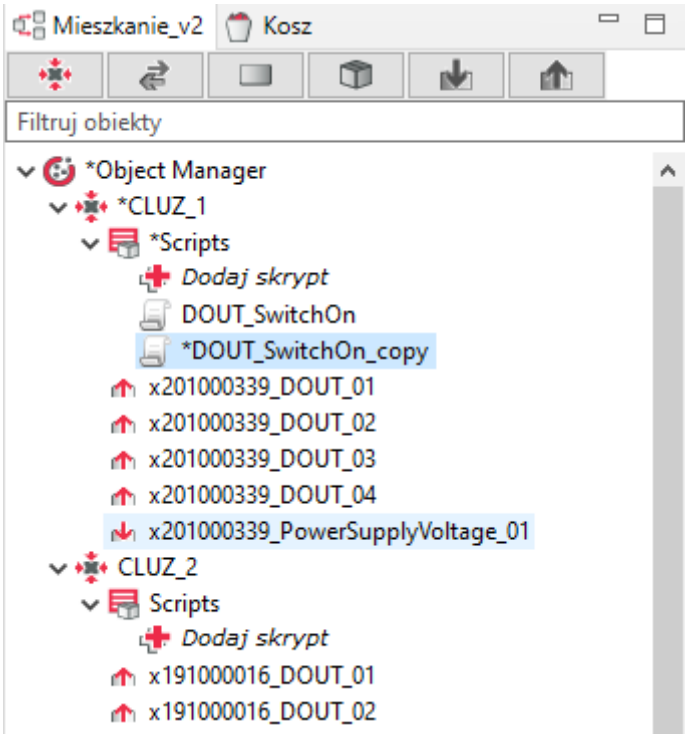
W przypadku skryptów tekstowych możliwe jest wykonanie kopii skryptu. Skrypt można skopiować w obszarze danego CLU jak również istnieje możliwość wykonania kopii skryptu do innego CLU.



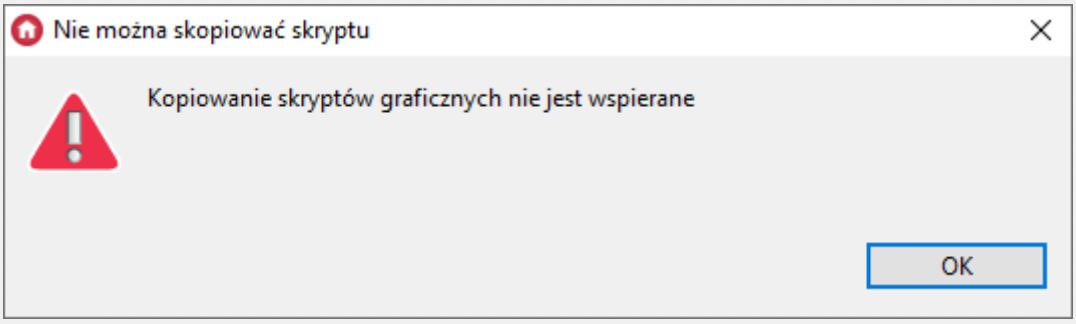
Po wywołowaniu opcji zostaje wyświetlone okno *Kopiowanie skryptu* w którym należy wybrać CLU do którego będzie kopiowany wybrany skrypt oraz należy podać nową nazwę skryptu.



Skopiowany skrypt zostaje dodany do listy skryptów danego (wybranego) CLU.



Uwaga! Kopiowanie skrytpów nie jest możliwe w przypadku skryptów graficznych. Przy próbie kopiowania zostanie wyświetlony komunikat:



3. Data i czas

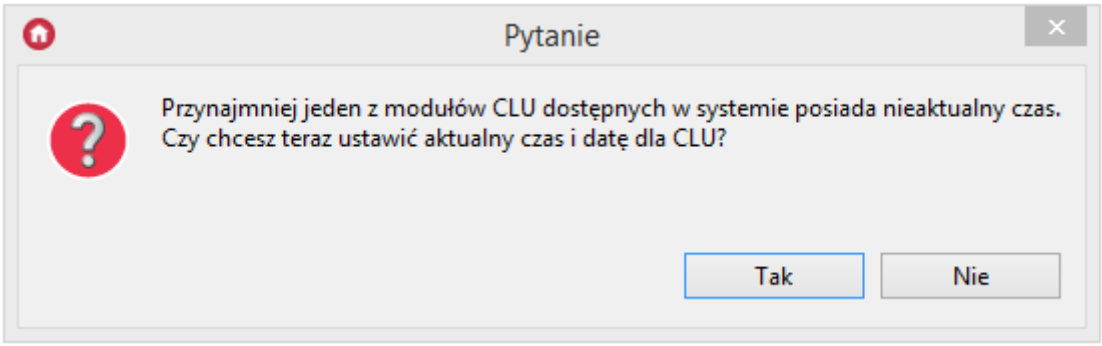
CLU wyposażone jest w zegar czasu rzeczywistego (RTC) podtrzymywany wbudowaną baterią. CLU udostępnia kilkanaście cech, które mogą być wykorzystywane w skryptach. Pełna lista cech związanych z czasem jest następująca:

Nazwa	Opis
`Uptime`	Czas pracy urządzenia od ostatniego resetu (w sekundach)
`Date`	Zwraca aktualną datę
`Time`	Zwraca aktualny czas (hh:mm:ss)
`Day`	Zwraca numer bieżącego dnia miesiąca
`Month`	Zwraca numer bieżącego miesiąca
`Year`	Zwraca numer bieżącego roku
`DayOfWeek`	Zwraca numer bieżącego dnia tygodnia (0=niedziela)
`Hour`	Zwraca aktualną godzinę (bez minut i sekund)
`Minute`	Zwraca aktualną liczbę minut od ostatniej pełnej godziny
`LocalTime`	Zwraca aktualny lokalny znacznik czasu
`TimeZone`	Zawiera aktualnie ustawioną strefę czasową
`NTPServer`	Zawiera ustawiony adres serwera czasu UTC

Warto zwrócić uwagę na cechę `LocalTime`, która podaje liczbę sekund od 1970 r. (czasu lokalnego) w postaci jednej liczby. Może być ona przydatna do sprawdzenia, ile czasu minęło od poprzedniego wykonania skryptu lub wywołania zdarzenia.

Aktualny czas (w postaci UTC) pobierany jest automatycznie do CLU z serwera NTP i korygowany przez ustawioną strefę czasową `TimeZone`. Można również ustawić go ręcznie przy pomocy metody `SetDateTime`.

Jeśli podczas otwierania projektu aplikacja Object Manager wykryje, że czas na CLU jest nieaktualny, zapyta użytkownika o automatyczne ustawienie aktualnego czasu i daty dla CLU.



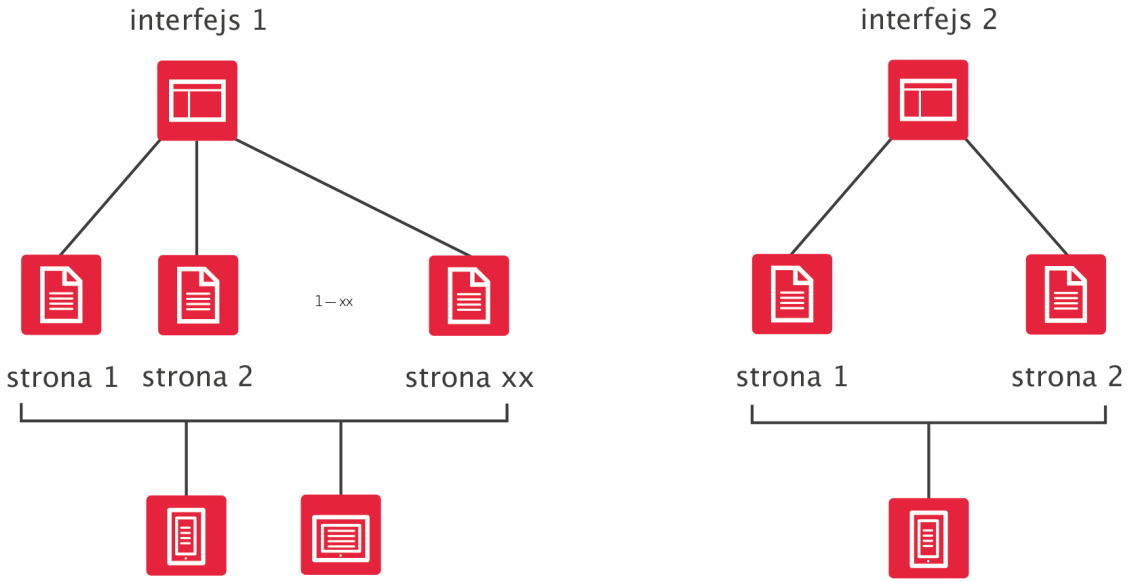
VIII. Visual Builder

> **Uwaga!** Wsparcie dla funkcjonalności Visual Builder zostało wstrzymane. Kreator interfejsów Home Manager zostanie usunięty w następnych wersjach Object Manager.

1. Sterowanie systemem z poziomu smartfonów

System umożliwia sterowanie przy pomocy dowolnych urządzeń pracujących w oparciu o system operacyjny Android oraz iOS (tablety, telefony komórkowe, media playery). Dla każdego systemu można przygotować wiele interfejsów użytkownika, z których każdy może zawierać sporą liczbę podstron. Umożliwia to stworzenie różnych interfejsów dla różnych użytkowników, zgodnie z ich potrzebami i upodobaniami, a także logiczne posortowanie funkcji sterujących w ramach każdego interfejsu (np. każde pomieszczenie na osobnej podstronie lub też podział wg funkcji - ogrzewanie, oświetlenie itp.).

Interfejsy są tworzone przy pomocy narzędzia Visual Builder będącego częścią Object Managera, a następnie przesyłane do aplikacji zainstalowanej na urządzeniu Android lub iOS.

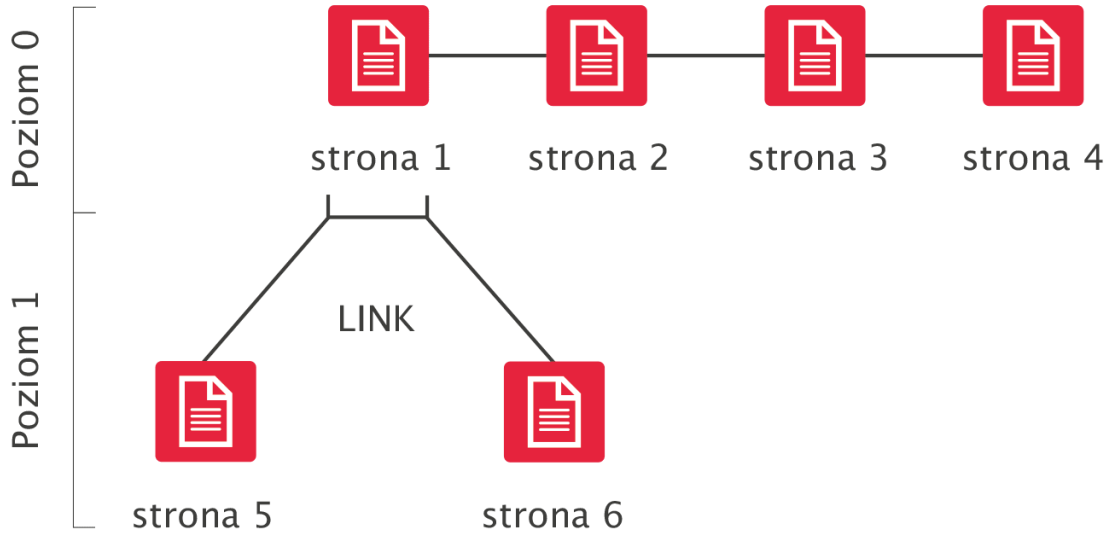


2. Struktura interfejsu

> **Uwaga!** Wsparcie dla funkcjonalności Visual Builder zostało wstrzymane. Kreator interfejsów Home Manager zostanie usunięty w następnych wersjach Object Manager.

Każdy interfejs składa się z jednej lub większej liczby podstron, na których rozmieszczane są elementy sterujące (przyciski, suwaki). Projektant może w pełni kontrolować układ stron, rozmieszczenie elementów graficznych oraz wygląd interfejsu, który ustalany jest poprzez zmianę skórki graficznej.

Strony w interfejsie mogą występować na dwóch poziomach: poziom zero oraz poziom jeden. Strony znajdujące się na poziomie zero dostępne są jako podstawowe strony interfejsu, po których nawigacja odbywa się poprzez przewijanie stron lewo/prawo. Do stron poziomu jeden, użytkownik może dostać się poprzez komponent *Link*.



3. Aplikacja na smartfony - GRENTON HOME MANAGER

Aplikacja GRENTON HOME MANAGER pozwala na uruchamianie interfejsów użytkownika zaprojektowanych w Visual Builderze na urządzeniach Android oraz iOS. Do aplikacji przesyłany jest gotowy pakiet przygotowany w Visual Builderze, zawierający opis interfejsu, wszystkie pliki z nim związane i dane konfiguracyjne.

W zależności od stworzonego interfejsu, aplikacja umożliwia przegląd bieżącego stanu systemu oraz sterowanie wszystkimi funkcjami dostępnymi w systemie.

W celu sterowania systemem GRENTON ze smartfona należy zainstalować w nim ww. aplikację, a następnie przesłać do niej interfejs stworzony przy pomocy Visual Buildera. Aplikację można bezpłatnie pobrać ze sklepu GOOGLE PLAY na urządzenia Android oraz z APP STORE na urządzenia iOS. Do poprawnego działania aplikacji, urządzenie, na którym została zainstalowana, musi być podłączone do tej samej sieci LAN, co system GRENTON lub musi zostać skonfigurowane połączenie zdalne.

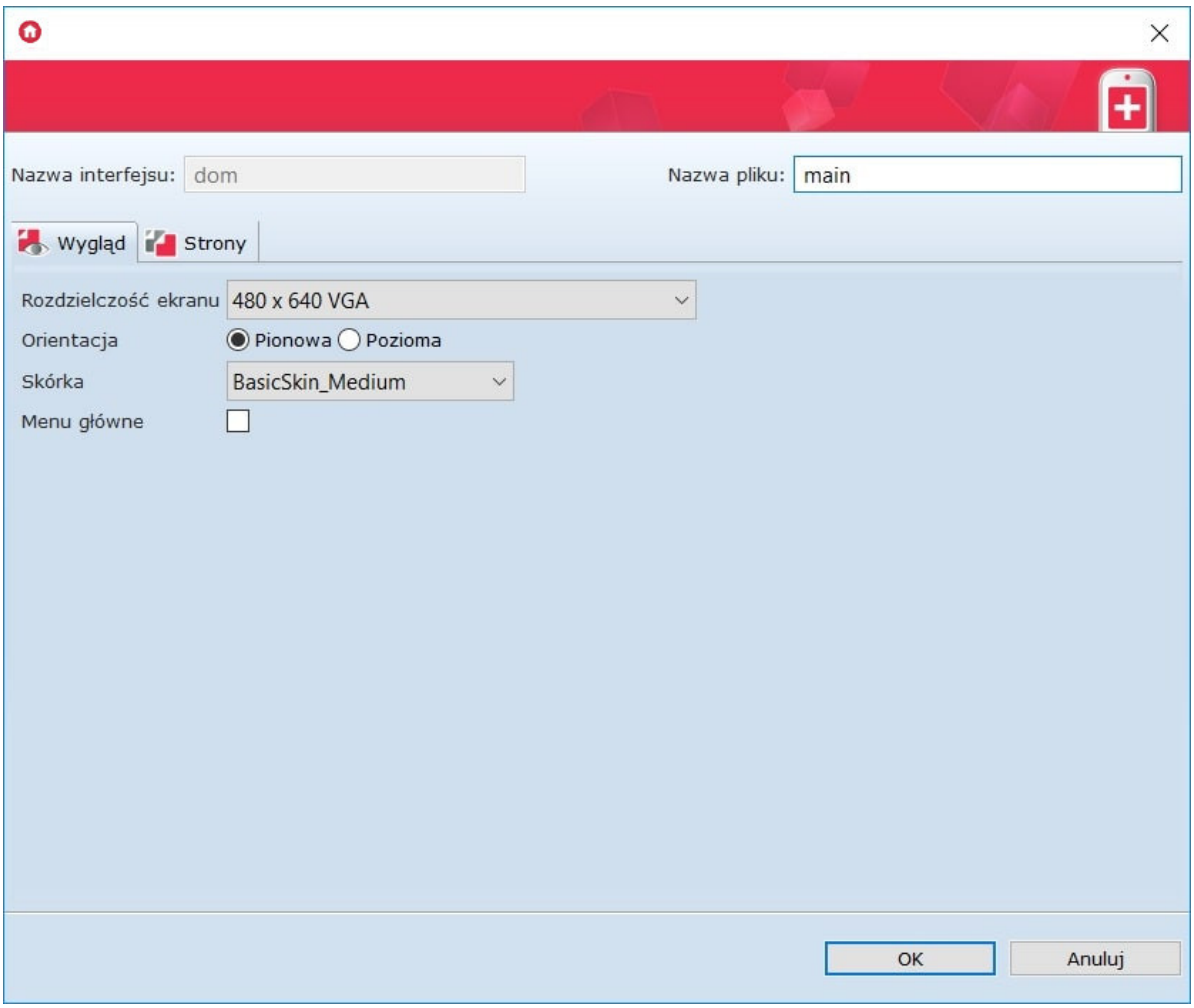
4. Tworzenie nowego interfejsu

> **Uwaga!** Wsparcie dla funkcjonalności Visual Builder zostało wstrzymane. Kreator interfejsów Home Manager zostanie usunięty w następnych wersjach Object Manager.

Utworzenie nowego interfejsu odbywa się poprzez wybranie ikony `Dodaj interfejs` z menu akcji.



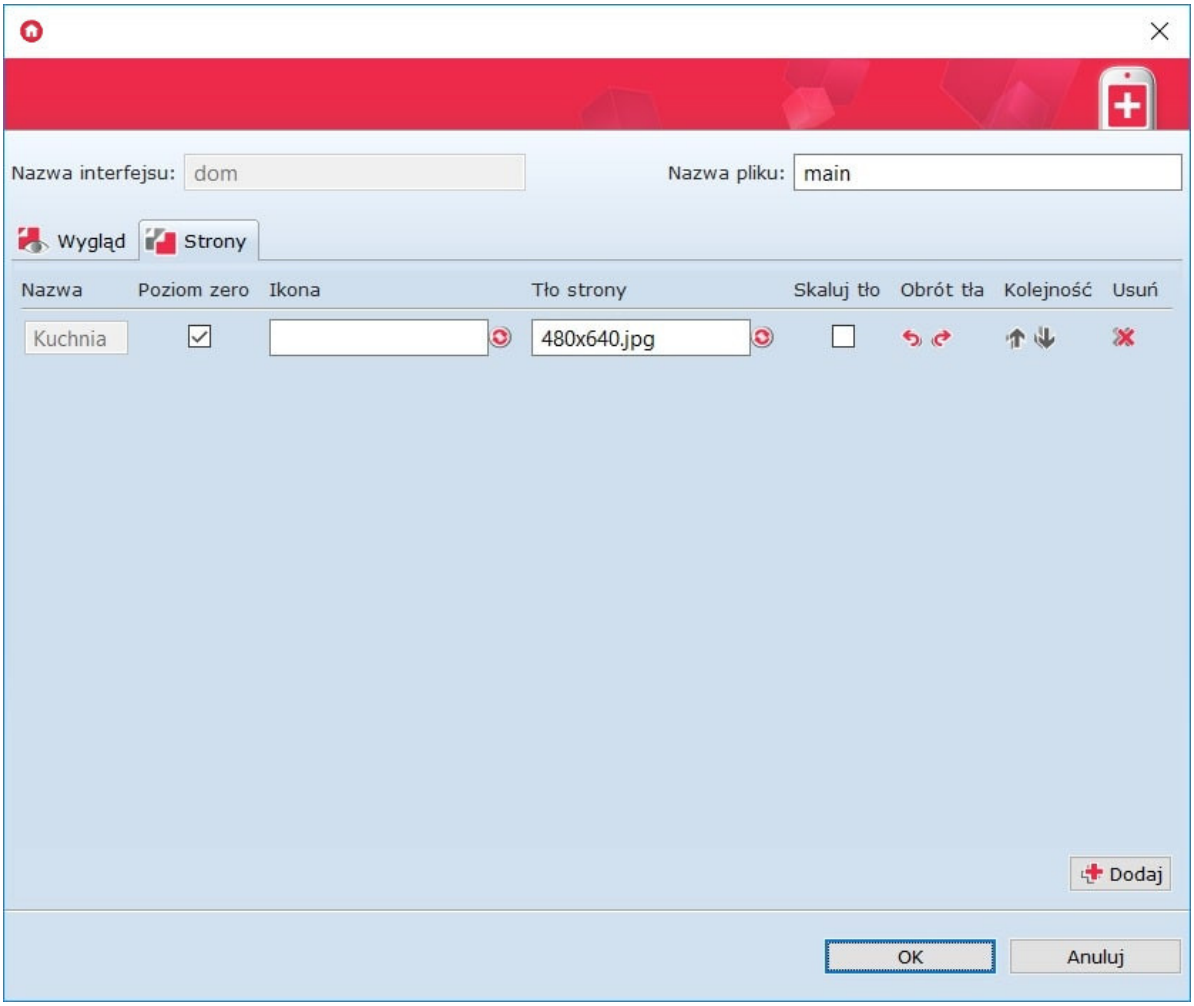
Po wpisaniu nazwy otworzy się okno utworzonego interfejsu, zawierające dwie zakładki: wygląd i strony (okno interfejsu jest również dostępne po dwukrotnym kliknięciu na ikonę utworzonego interfejsu w menu obiektów). Zakładka `Wygląd`:



Zawiera informacje dotyczące sposobu wyświetlania interfejsu, takie jak: rozdzielczość, orientacja, lista dostępnych skórek oraz pole, po zaznaczeniu którego tworzone jest menu główne.

W prawym górnym rogu znajduje się pole `Nazwa pliku`. Nazwa ta, po wysłaniu interfejsu do urządzenia mobilnego, jest wyświetlana na jego liście interfejsów. W przypadku, gdy do jednego urządzenia wysyłany jest więcej niż jeden interfejs, należy pamiętać o przypisaniu każdemu z nich innej nazwy.

W zakładce `Strony` zawarta jest lista wszystkich utworzonych stron.



W tej zakładce można dokonać zmiany kolejności wyświetlania się stron oraz usunąć wcześniej utworzone. Po zaznaczeniu opcji `Poziom zero` strona będzie widoczna w menu głównym. Dodatkowo, w zakładce *Strony*, możliwa jest zmiana ikony, wyświetlanej u dołu strony w menu oraz jej tło.

Jeśli wybrane tło ma orientację inną niż używana w interfejsie, wolno dokonać jego obrotu za pomocą przycisków `Obrót tła`.

Dodatkowo istnieje możliwość skalowania tła. Wybór tej opcji powoduje dopasowanie dowolnej rozdzielczości tła do rozdzielczości tworzonego interfejsu.

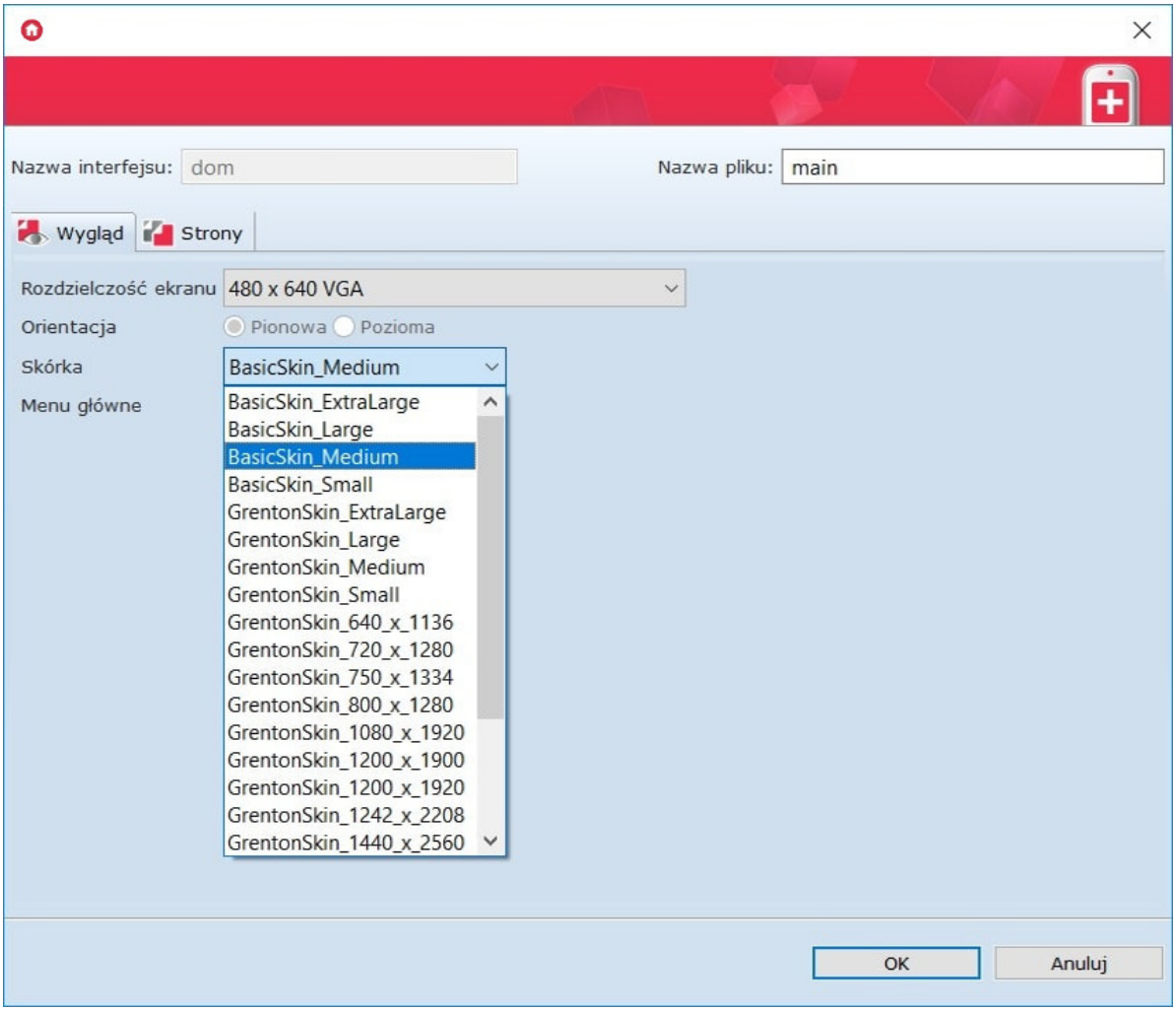
> **Uwaga!** Nowo utworzony projekt w zakładce *Strony* nie posiada żadnych informacji. Informacje pojawiają się tam w momencie utworzenia stron interfejsu.

4.1. Wybór skórki graficznej

Skórki to zestaw ustawień graficznych dla interfejsu na urządzenie mobilne.

Skórki GRENTON

Użytkownik, w tworzonych interfejsach, może wykorzystać dostarczone wraz z OM skórki. Lista dostępnych skórek znajduje się w parametrach interfejsu mobilnego.



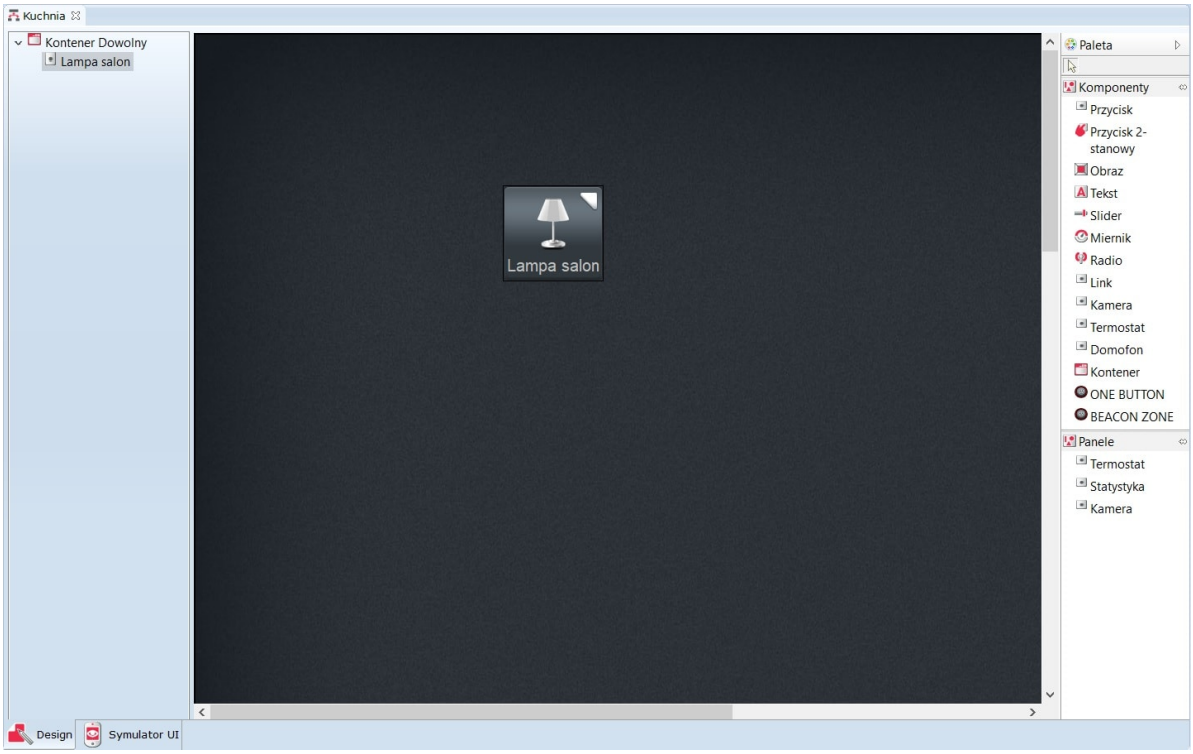
4.2. Tworzenie stron interfejsu

Po utworzeniu interfejsu, należy dodać do niego nowe strony. Tworzenie strony odbywa się z menu akcji:



Po utworzeniu nowej strony oraz nadaniu jej nazwy zostaje otwarty arkusz edycji. Znajdują się w nim dwie zakładki: *Design* oraz *Symulator UI* (zakładki znajdują się u dołu strony).

W zakładce *Design* znajdują się: lista obiektów, kontener główny, a także lista komponentów i paneli.



Lista obiektów wyświetla wszystkie obiekty wykorzystane w bieżącym arkuszu roboczym.

4.3. Komponenty

Komponenty - lista obiektów, które mogą być wykorzystane podczas tworzenia interfejsu. Lista komponentów obejmuje:

- **Przycisk** - działa jak przycisk monostabilny.

  Przycisk

- **Przycisk** - działa jak przycisk bistabilny.

  Przycisk 2-stanowy

- **Obraz** - umożliwia wstawienie obrazka z pliku zewnętrznego.

  Obraz

- **Tekst** - umożliwia wstawienie pola tekstowego.

  Tekst

- **Slider** - suwak umożliwiający wykonanie płynnej regulacji.

  Slider

- **Miernik** - odwzorowuje w sposób analogowy wartość obiektu.



- **Radio** – odwzorowuje w sposób cyfrowy (on/off) stan danego obiektu.



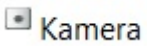
- **Link** – umożliwia tworzenie linków do innych stron w ramach tego samego interfejsu.



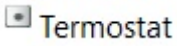
- **Kontener** – porządkuje układ komponentów na obszarze roboczym w określony sposób.



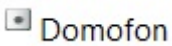
- **Kamera** – umożliwia wyświetlanie obrazu z kamery IP w aplikacji Home Manager.



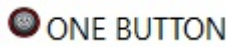
- **Termostat** – umożliwia wyświetlenie obiektu wirtualnego Termostat w aplikacji Home Manager.



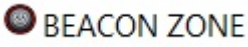
- **Domofon** – umożliwia skonfigurowanie domofonu (skonfigurowanie połączenia z serwerem *SIP*, przypisanie metod do konkretnych zdarzeń oraz wyświetlanie obrazu z kamery IP podczas połączenia).



- **ONE BUTTON** – umożliwia przypisanie BEACONOWI metody do zdarzenia w trybie ONE BUTTON.



- **BEACON ZONE** – umożliwia skonfigurowanie BEACONA w trybie BEACON ZONE oraz przypisanie konkretnych metod do zdarzeń (po dodaniu do strony widoczny u jej dołu pasek BEACON ZONE).



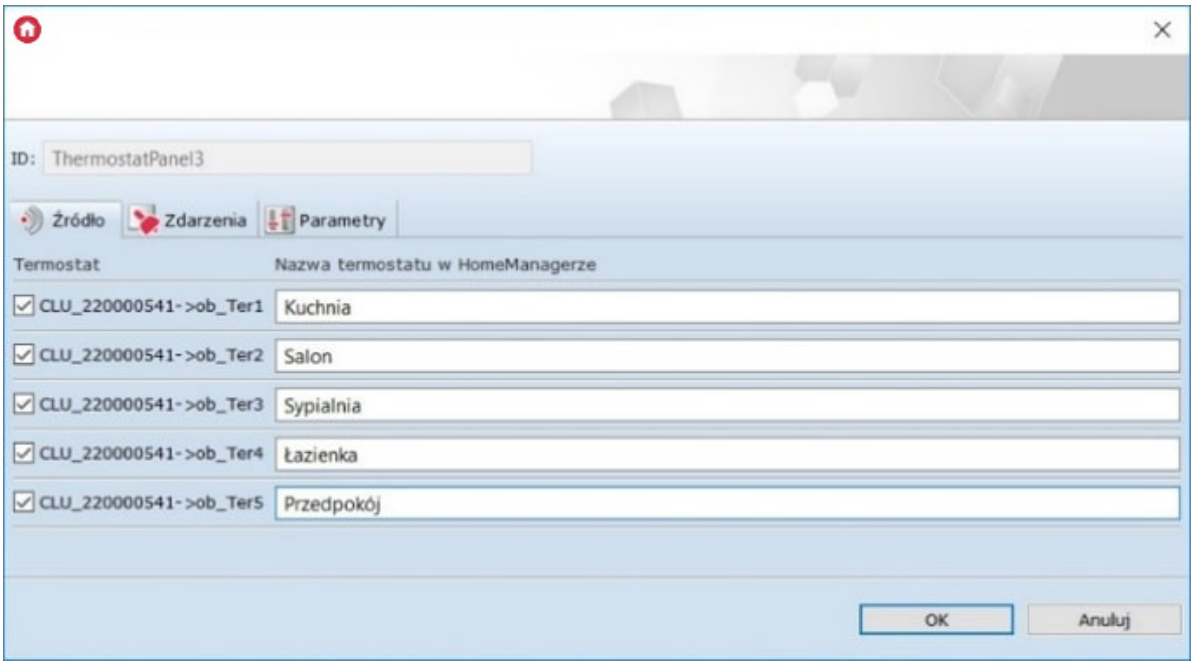
Wybrane obiekty wstawiane są do kontenera z listy komponentów poprzez przeciąganie, a ich rozmieszczenie jest uzależnione od typu zastosowanego kontenera głównego.

4.4. Panele

Panele – lista obiektów, które mogą zostać wykorzystane podczas tworzenia interfejsu na urządzenie mobilne. Panele w odróżnieniu do komponentów zajmują całą stronę interfejsu mobilnego. Lista paneli obejmuje:

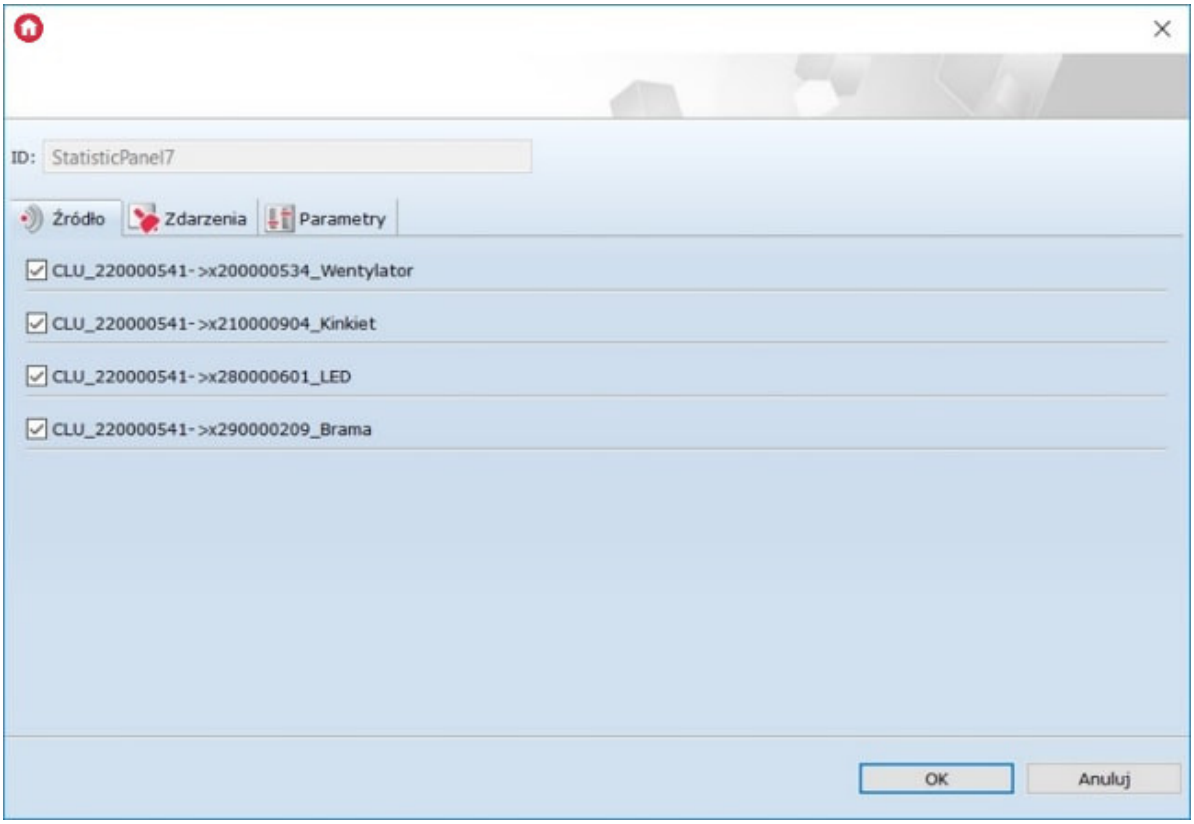
- **Termostat** – tworzy panel dla termostatu na całej stronie interfejsu w HM. Termostat

Jako źródło panelu termostatu ustawia się wcześniej utworzony obiekt wirtualny `Termostat`.



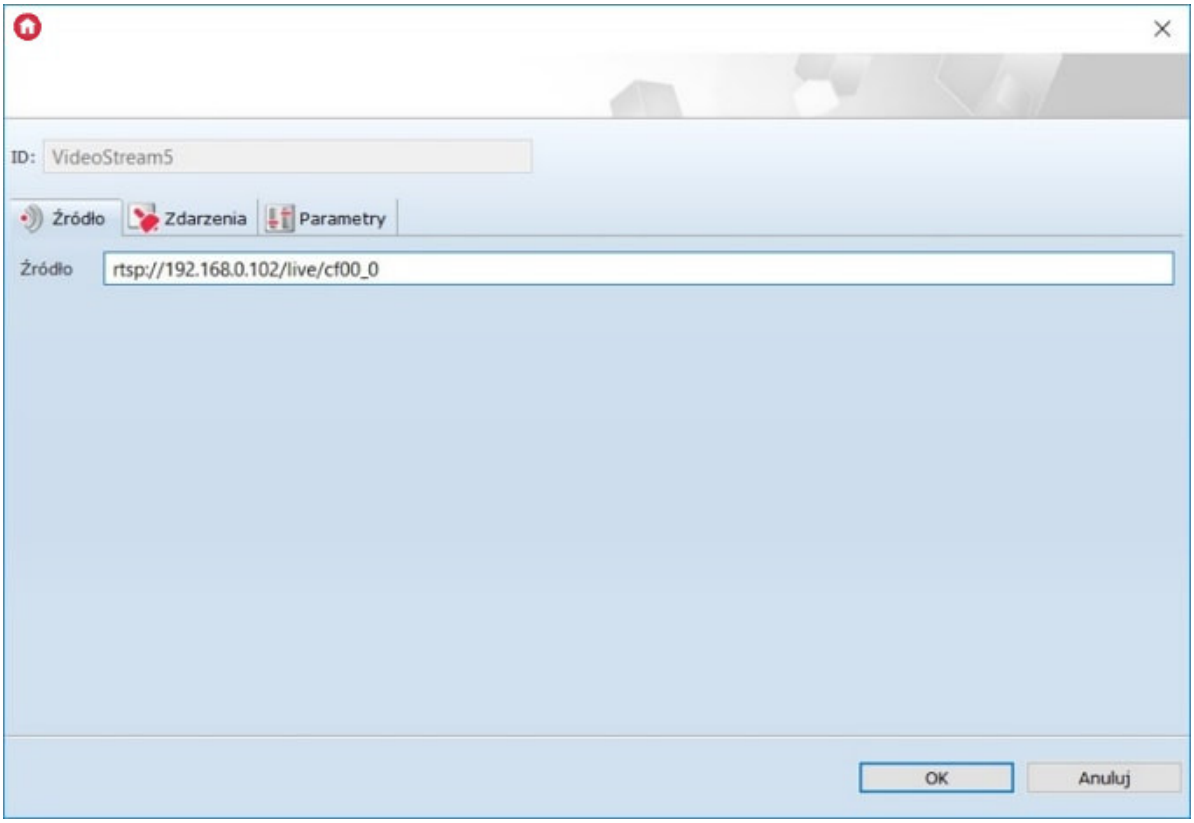
- **Statystyka** – tworzy panel do pomiaru mediów na stronie interfejsu w HM. Statystyka

Po przeciągnięciu panelu na stronę interfejsu należy wybrać obiekty, dla których pomiar mediów będzie prezentowany w HM. W oknie wyświetlone będą jedynie obiekty, dla których wcześniej załączono *Pomiar Mediów*.



- **Kamera** – tworzy panel do wyświetlania obrazu z kamery IP na zdefiniowanej przestrzeni strony interfejsu w HM. Kamera

Jako źródło panelu kamery należy podać strumień RTSP danej kamery IP.

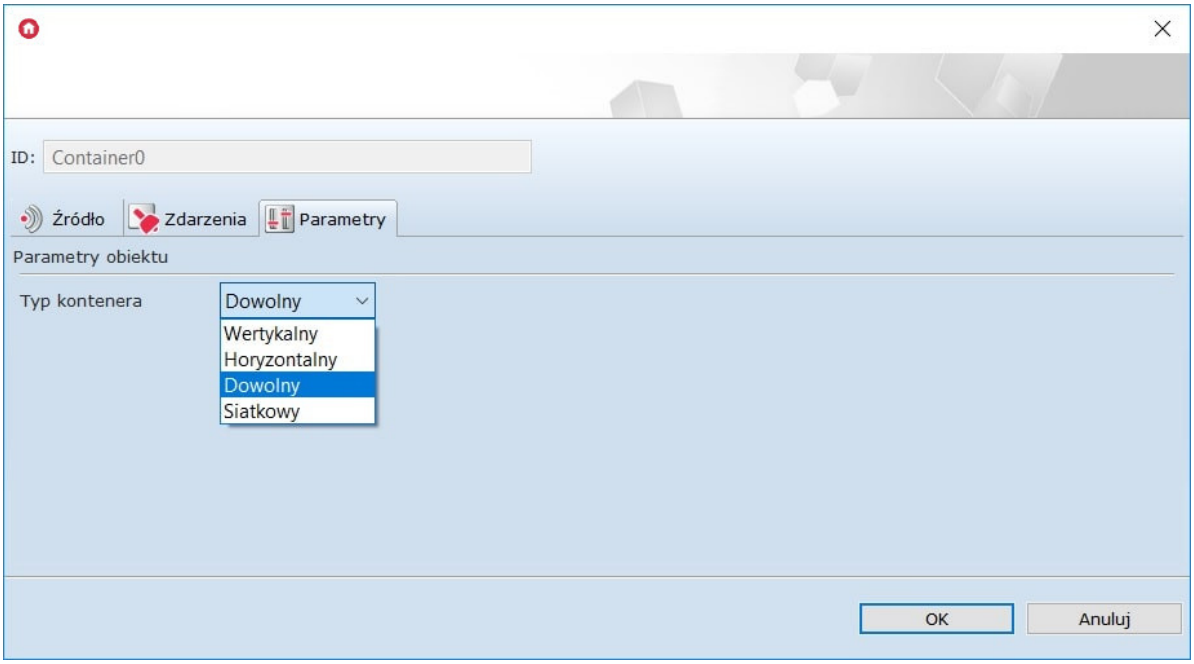


4.5. Kontenery

Kontener to zasobnik obiektów, określający ich sposób rozmieszczenia w obszarze roboczym.

Obiekty wewnątrz obszaru roboczego rozmieszczane są zgodnie z typem wybranego kontenera.

Typ kontenera można zmienić w parametrach obiektu dla tego kontenera. Okno z parametrami otwiera się po dwukrotnym kliknięciu na obiekt kontenera, znajdujący się na pierwszym miejscu listy obiektów.

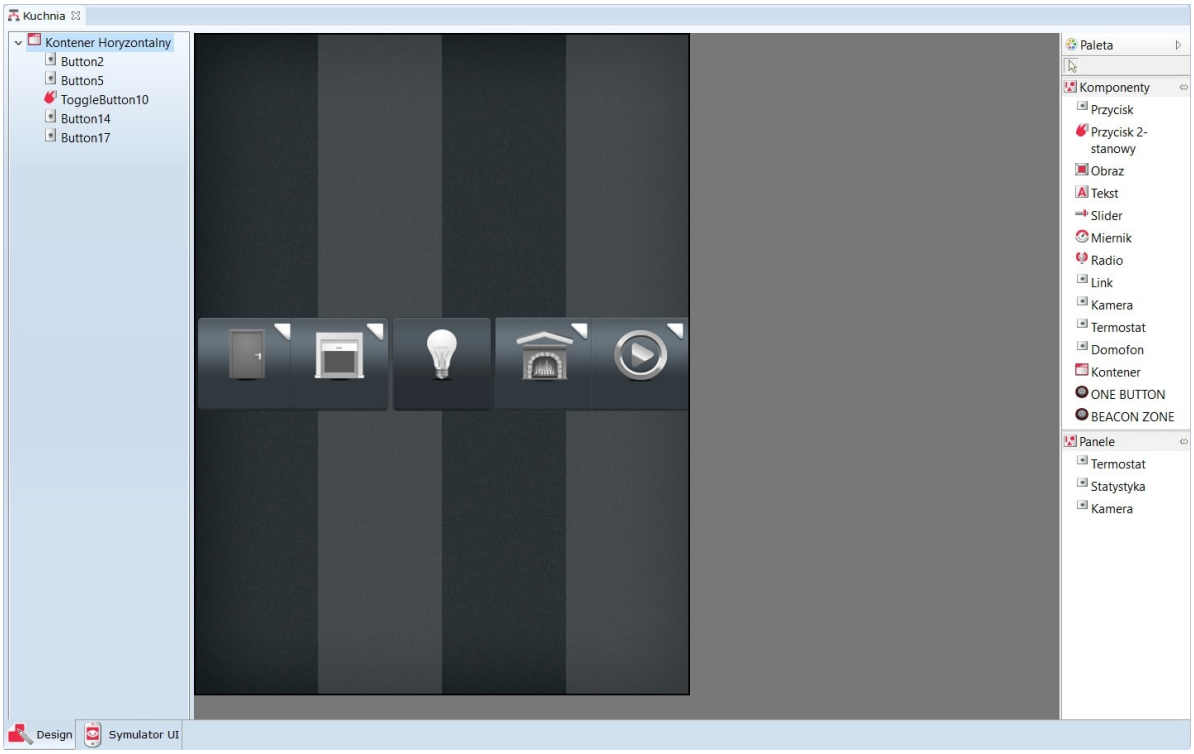


Istnieją cztery typy dostępnych kontenerów:

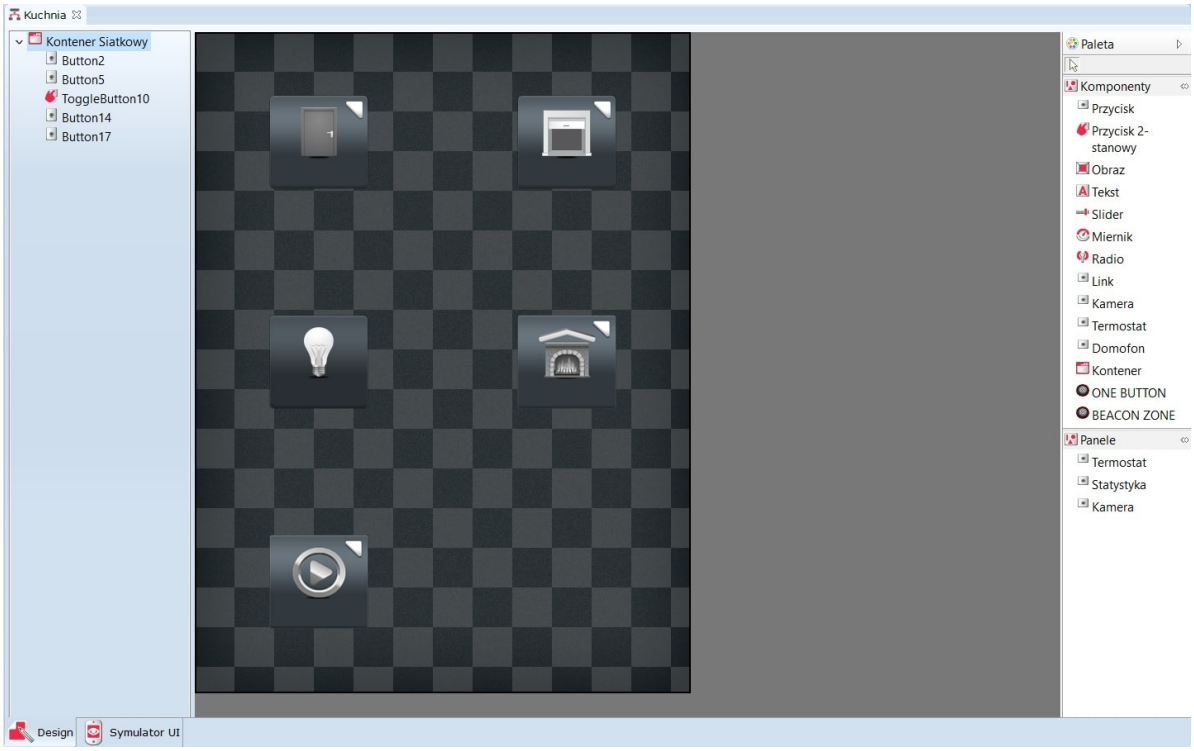
1. **Wertykalny** – rozmieszcza wstawiane elementy w sposób pionowy w równych, automatycznie się tworzących sekcjach.



2. **Horyzontalny** – wstawiane elementy rozmieszczane są w poziomych sekcjach.



3. **Siatkowy** - wstawiane komponenty układane są wg symetrycznej siatki.



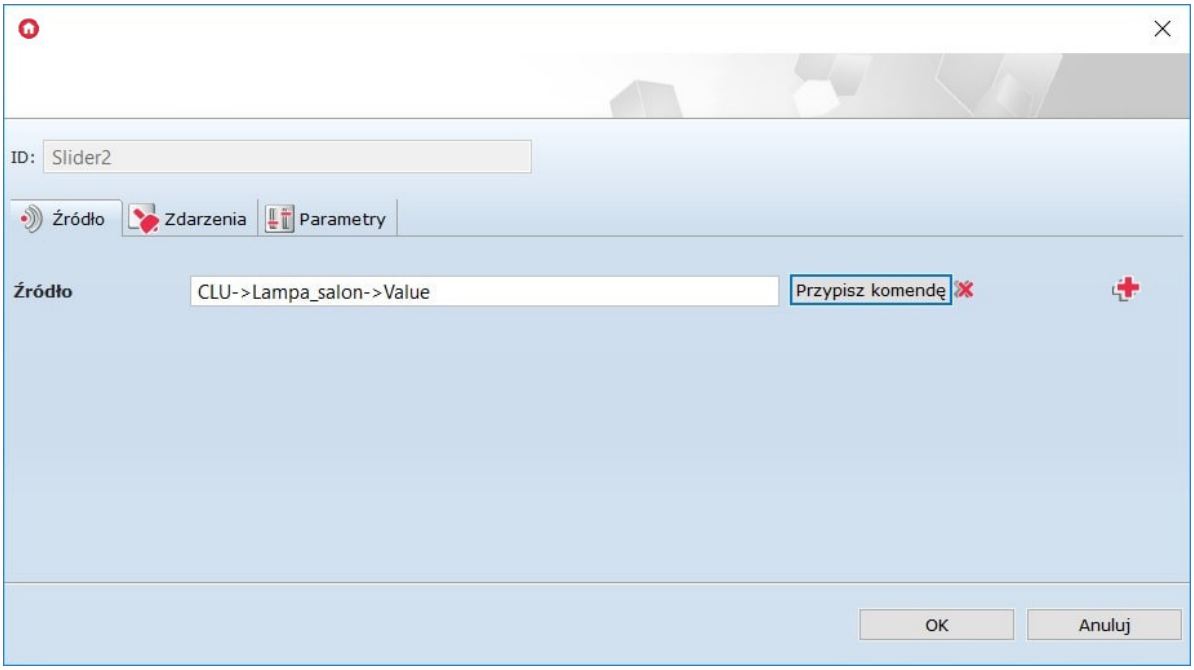
4. **Dowolny** - umożliwia dowolny rozkład wstawionych komponentów na całym obszarze kontenera.



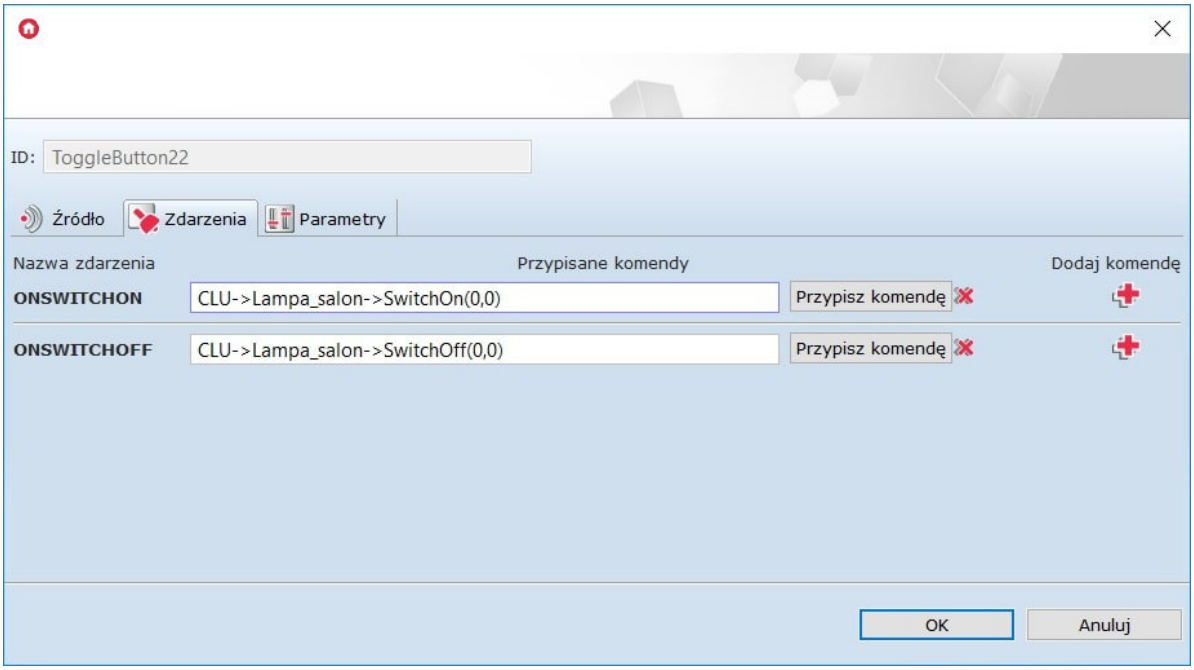
4.6. Wstawianie komponentów i tworzenie powiązań z obiektami systemu

Po wybraniu komponentu z listy znajdującej się po prawej stronie i wstawieniu go do kontenera głównego, automatycznie otwiera się okno właściwości dla tego obiektu. W oknie tym znajdują się trzy zakładki (`Źródło` , `Zdarzenia` oraz `Parametry`), w których należy ustawić odpowiednio:

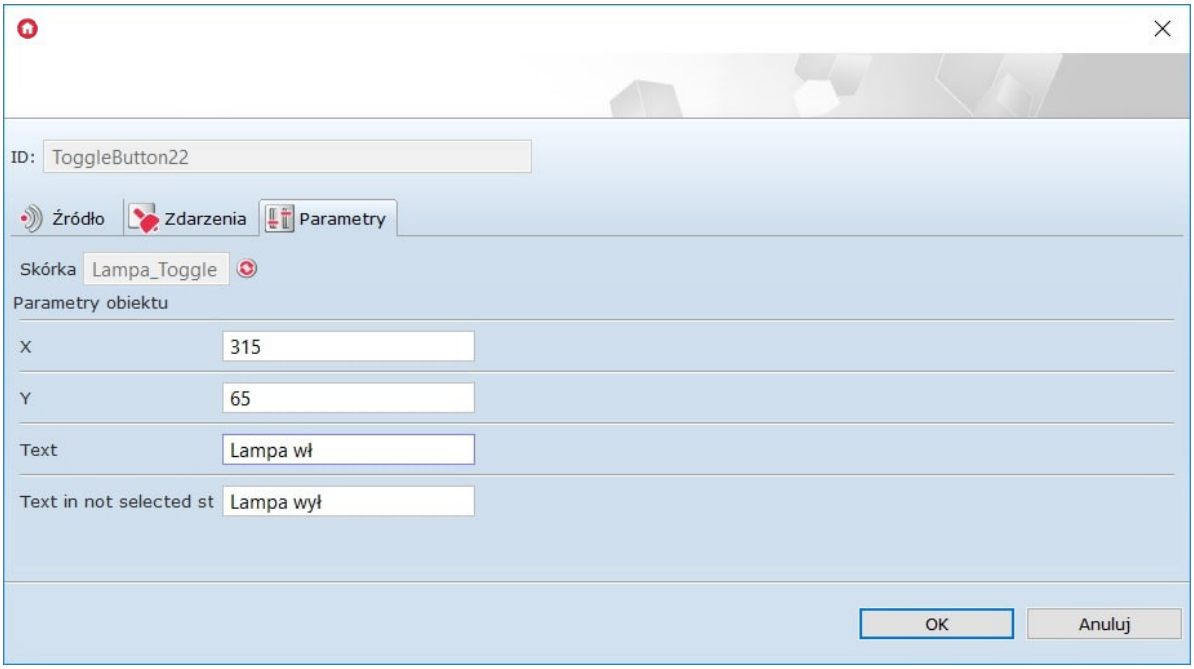
1. W zakładce `Źródło` należy wskazać obiekt, którego wartość ma być odwzorowana oraz czas odświeżania dla tej wartości, np. jeżeli wstawimy do interfejsu slider, który ma sterować dimmerem, to w celu wyświetlania na smartfonie aktualnej wartości załączonego oświetlenia, jako źródło musi zostać wskazany sterowany dimmer.



2. `Zdarzenia` wykorzystywane są dla obiektów, za pomocą których odbywa się sterowanie, np. przycisk lub slider. W tej zakładce znajdują się zdarzenia odpowiednie dla danego typu obiektów, do których należy przypisać metody sterowanych obiektów.



3. W zakładce `Parametry` znajdują się dane związane z wyświetlaniem danego obiektu w interfejsie. Użytkownik ma możliwość: zmiany skórki obiektu, ustawienia położenia komponentu na stroni, a także edycji wyświetlanego podpisu obiektu.



> **Uwaga!** Jeśli w polu `Text` wprowadzona zostanie komenda ***$value$***, to będzie w nim wyświetlana aktualna wartość cechy `Value` obiektu wybranego w zakładce `Źródło`.

Po ukończeniu komponowania interfejsu lub w trakcie jego tworzenia, użytkownik ma możliwość sprawdzania działania oraz wyglądu przygotowanego interfejsu. W tym celu należy uruchomić Symulator UI (druga zakładka na dole strony).

4.7. Wysyłanie interfejsu do urządzenia mobilnego

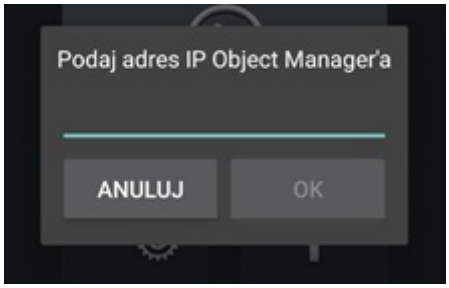
Chcąc sterować systemem za pomocą urządzenia mobilnego, utworzony interfejs musi zostać wysłany do aplikacji GRENTON HOME MANAGER zainstalowanej na wybranym urządzeniu.

W tym celu należy:

- Wybrać z listy utworzonych interfejsów Visual Buildera ten, który ma zostać wysłany - ikona narzędzia do wysyłania interfejsu znajduje się w menu głównym:



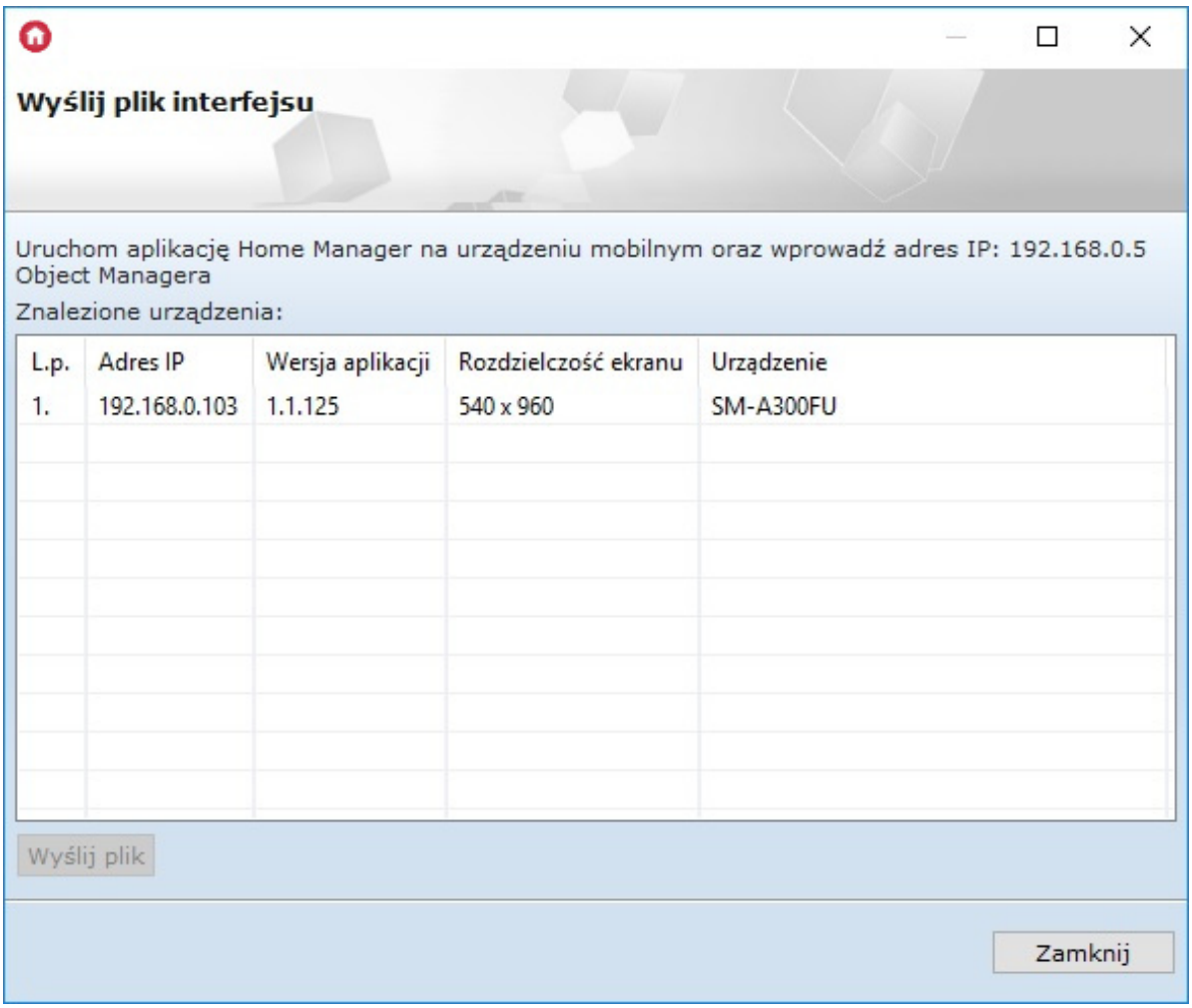
- W urządzeniu mobilnym połączyć się z siecią, w której znajduje się CLU (po wyświetleniu w Object Managerze okna wysyłania);
- W otwartej aplikacji Home Manager wybrać z menu głównego *Połącz z OM*;
- Podać adres IP Object Managera i wybrać *OK*:



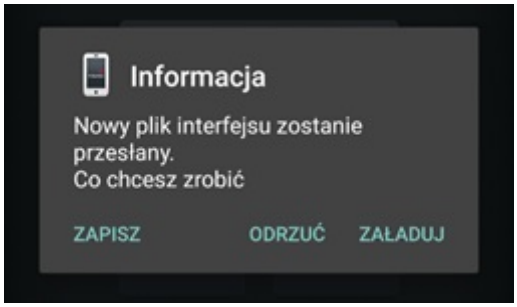
- Urządzenie mobilne pojawi się w oknie wysyłania, które zostało wyświetlone w programie Object Manager;

> **Uwaga!** Na liście wyświetlane są urządzenia, na których uruchomiona jest aplikacja GRENTON HOME MANAGER i została w niej włączona opcja `Połącz z OM` w ustawieniach aplikacji.

- Kliknąć myszką dwukrotnie na jego nazwę lub zaznaczyć i wybrać *Wyślij plik*:



- W aplikacji mobilnej pojawi się okno przyjmowania interfejsu. Wybrać *Zapisz*:



- Na ekranie pojawi się pasek stanu przesyłania. Po zakończeniu, na górnej belce programu, wyświetli się informacja o poprawnym zakończeniu przesyłania.
- Po wysłaniu pliku z interfejsem do urządzenia mobilnego, by zdalne sterowanie było możliwe, należy załadować przesłany interfejs.

5. Automatyczne tworzenie interfejsu - generator GUI

> **Uwaga!** Wsparcie dla funkcjonalności Visual Builder zostało wstrzymane. Kreator interfejsów Home Manager zostanie usunięty w następnych wersjach Object Manager.

Funkcja ta umożliwia szybkie stworzenie interfejsu poprzez wybór obiektów, którymi chcemy sterować spośród wszystkich obiektów dostępnych w systemie.

Automatyczne tworzenie interfejsu użytkownika rozpoczynamy od uruchomienia Generatora GUI. Ikona generatora znajduje się w menu obiektów:

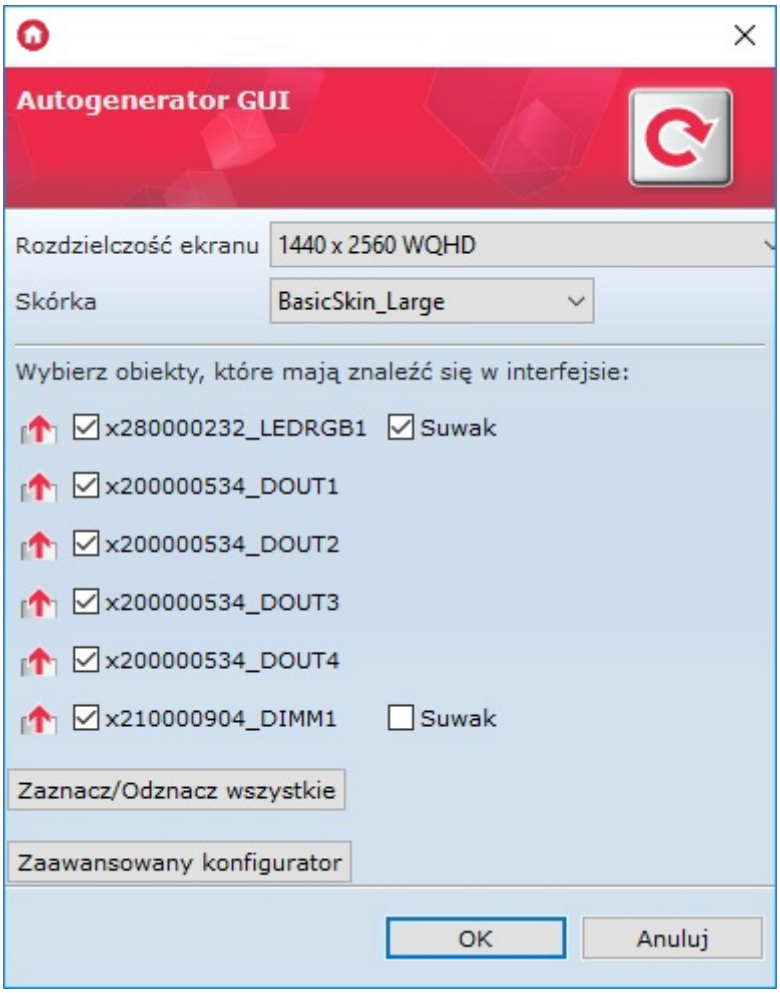


5.1. Tworzenie interfejsu z dostępną rozdzielczością

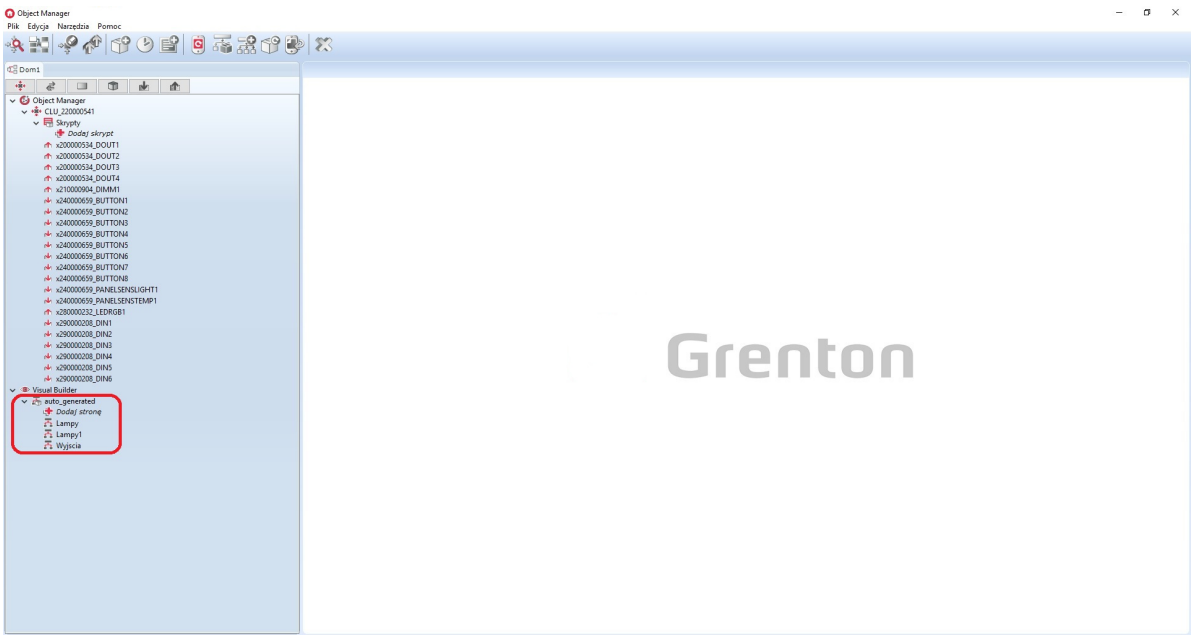
A. Konfigurator prosty

Po kliknięciu na wskazaną ikonę otwiera się okno `Autogeneratora GUI`. Jest to prosty konfigurator, w którym należy wybrać:

- rozdzielczość, z jaką pracuje urządzenie mobilne;
- skórkę określającą wygląd ikon w interfejsie;
- obiekty (z listy obiektów), które mają zostać zawarte w utworzonym interfejsie.

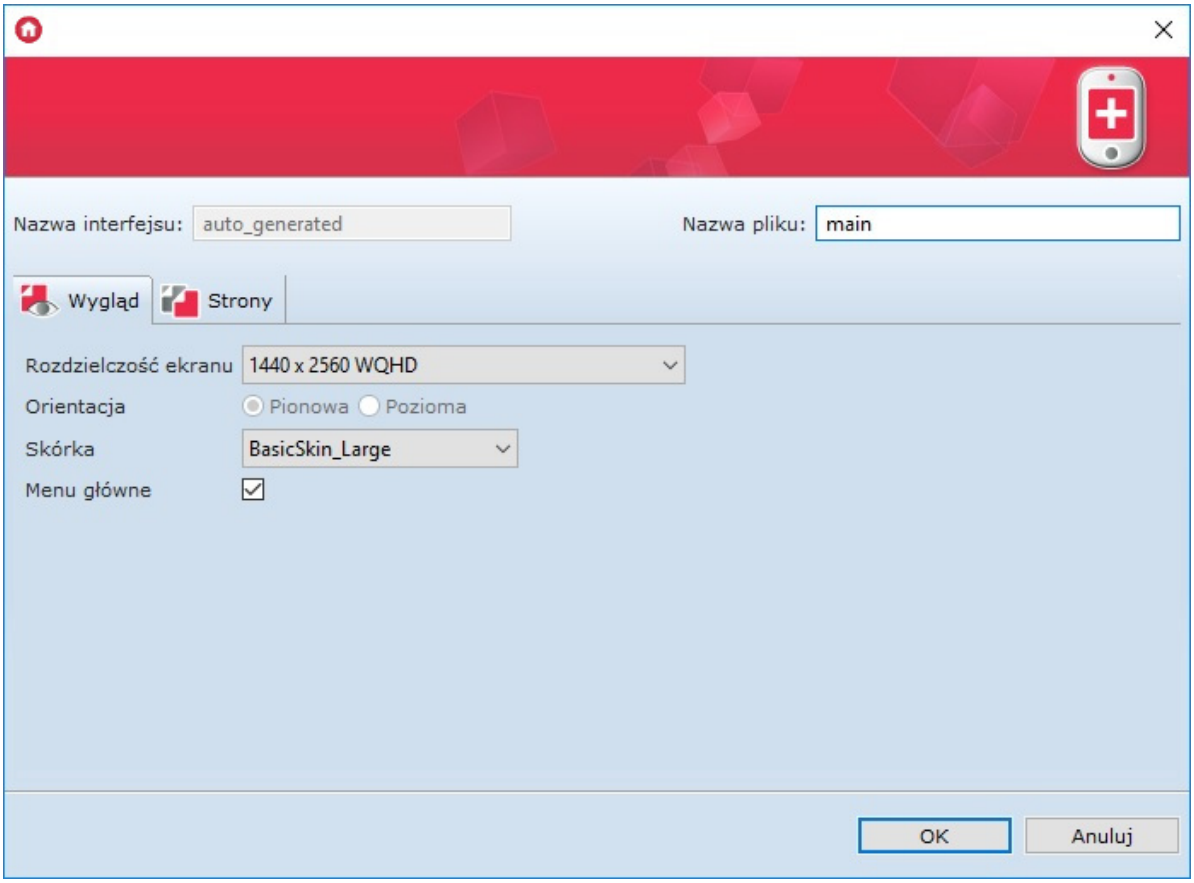


Po zaznaczeniu interesujących obiektów należy kliknąć `OK`. W rezultacie na liście obiektów (pod ikoną utworzonego interfejsu) pojawiają się nowo utworzone strony zgodnie z poniższym rysunkiem:



W każdej chwili istnieje możliwość zmiany ustawień interfejsu - wystarczy dwukrotnie kliknąć na jego nazwę, a otworzy się okno z dwoma zakładkami: `Wygląd` i `Strony`.

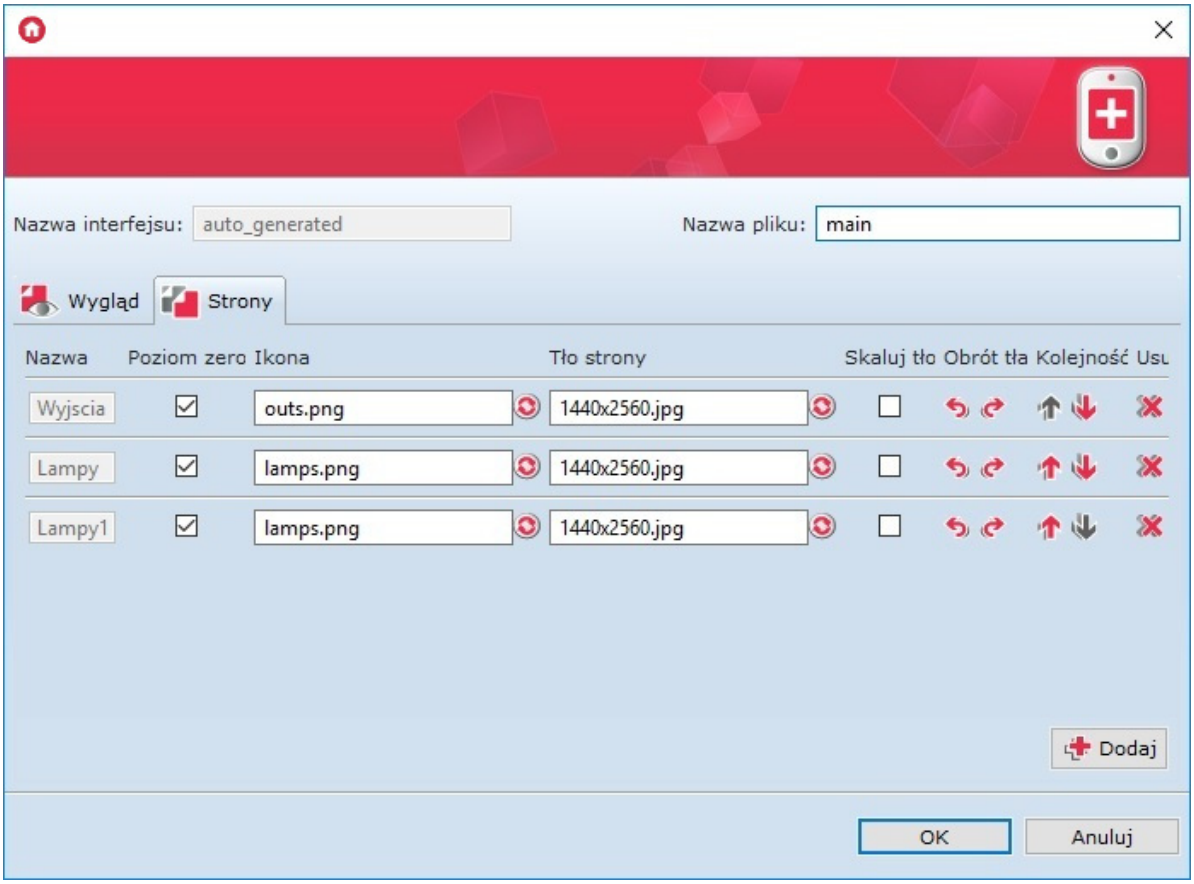
W zakładce `Wygląd` użytkownik ma możliwość wybrania skórek widocznych w interfejsie. W tym widoku znajduje się również pole *Menu główne*. Po jego zaznaczeniu zostanie utworzone menu, zawierające wszystkie dostępne i zaznaczone strony.



Zakładka `Strony` zawiera listę utworzonych stron oraz umożliwia zmianę ich parametrów, takich jak:

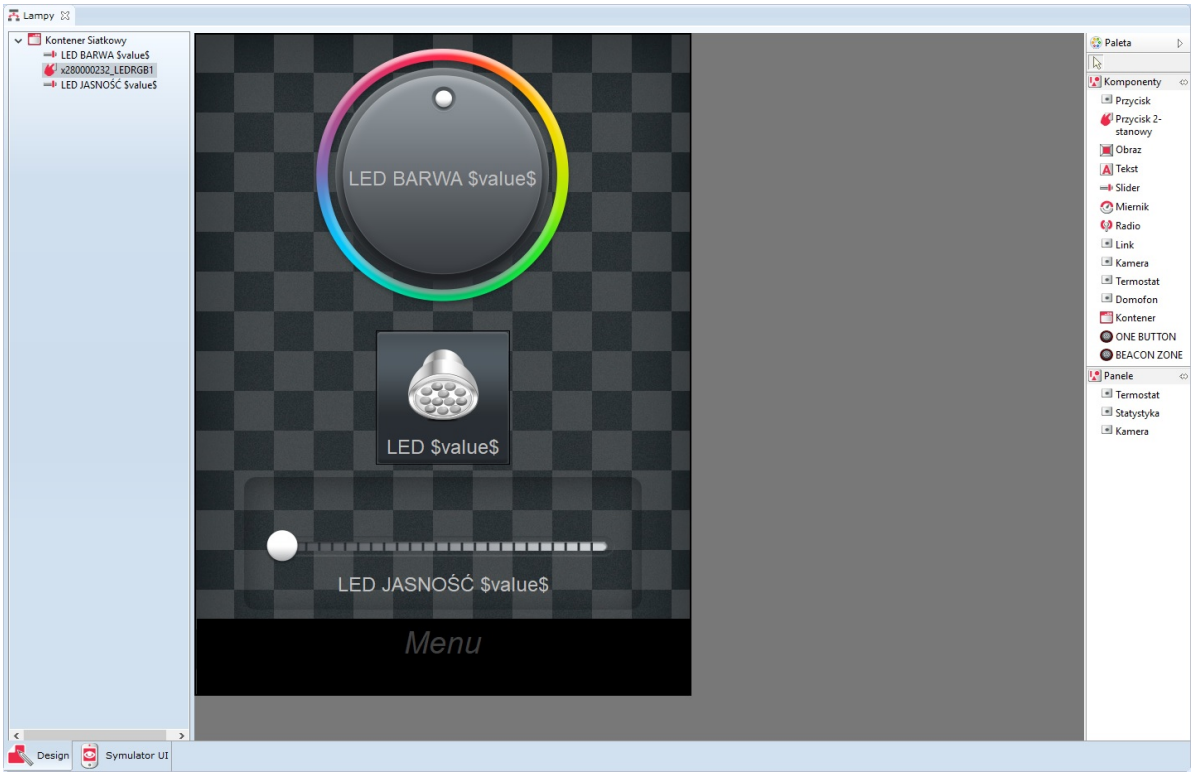
- **poziom zero** – określenie, czy strona ma być wyświetlana w menu;
- **ikona** – ikona wyświetlana w menu (domyślnie jest to ikona zgodna z wybraną skórką);
- **tło strony** – tło wyświetlanej strony (domyślnie wyświetlane jest tło ze skórki, jednakże użytkownik może zdefiniować własne);
- **skaluj tło** - dopasowanie wybranej rozdzielczości do rozdzielczości urządzenia mobilnego;

- **obrót tła** - zmiana orientacji tła;
- **kolejność** – ustawienie kolejności wyświetlania stron w menu;
- **usuń** – oznacza całkowite usunięcie strony z interfejsu.

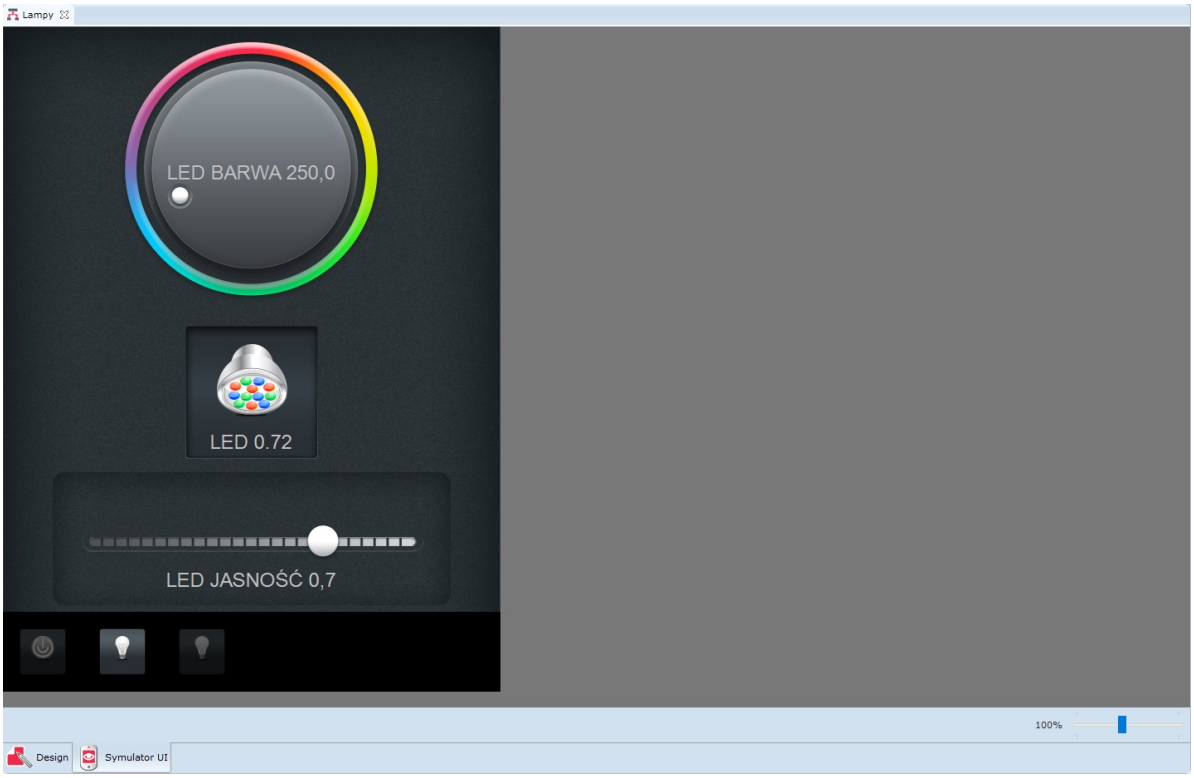


Użytkownik ma również możliwość dokonywania zmian w wygenerowanych stronach. Dwukrotne kliknięcie na ikonie strony otworzy arkusz edycji, zawierający dwie zakładki `Design` oraz `Symulator`.

Zakładka `Design` - wyświetla obszar roboczy zawarty w kontenerze oraz umożliwia edycję utworzonej strony.



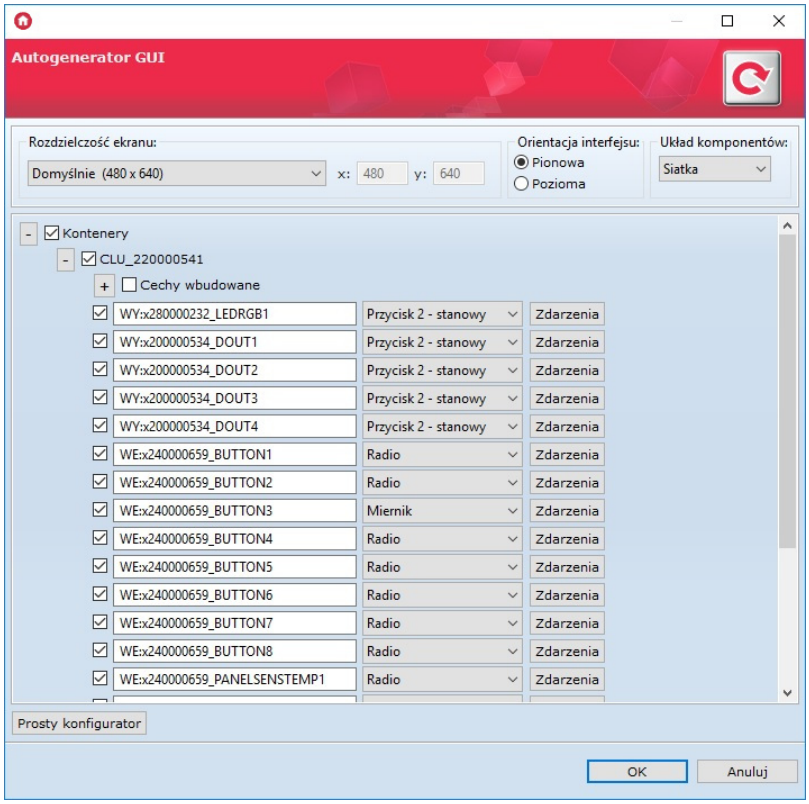
Zakładka `Symulator` - daje użytkownikowi możliwość sprawdzenia wyglądu oraz sposobu działania utworzonego interfejsu z ekranu komputera (zanim zostanie wysłane do urządzenia mobilnego).



B. Konfigurator zaawansowany

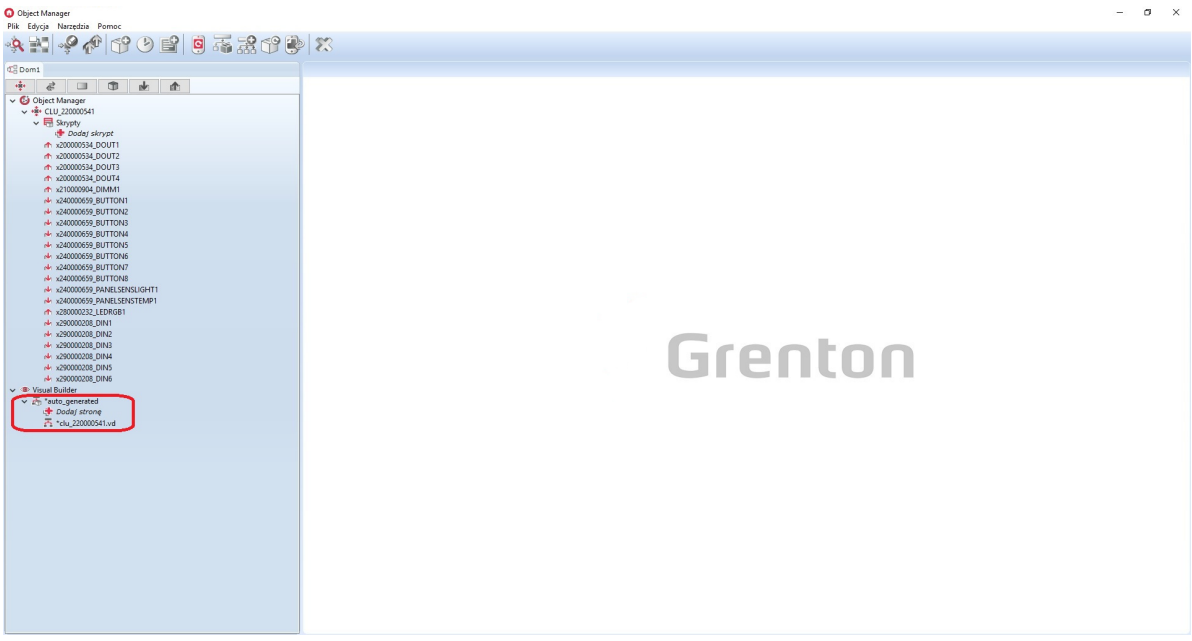
Po kliknięciu ikony `Generuj GUI` w oknie `Autogeneratora` istnieje możliwość wybrania opcji `Zaawansowany konfigurator`. Wybór tej opcji spowoduje otwarcie nowego okna, w którym należy wybrać:

- rozdzielczość, z jaką pracuje rządzenie mobilne;
- orientację interfejsu (pionową lub poziomą);
- układ komponentów (siatka lub lista);
- obiekty i cechy (z listy obiektów), które mają zostać zawarte w utworzonym interfejsie;
- wyświetlaną ikonę oraz zdarzenia dla każdego obiektu.



Następnie, po ustawieniu wszystkich parametrów i naciśnięciu `OK`, otwiera się okno utworzonego interfejsu. Okno to, poza polem nazwy utworzonego interfejsu, zawiera dwie zakładki: `Wygląd` i `Strony`. Ich funkcjonalności są dokładnie takie same, jak w przypadku konfiguratora prostego.

Po ustawieniu wszystkich parametrów w oknie utworzonego interfejsu i kliknięciu `OK`, na liście obiektów (pod ikoną utworzonego interfejsu) pojawiają się nowo utworzone strony zgodnie z poniższym rysunkiem:

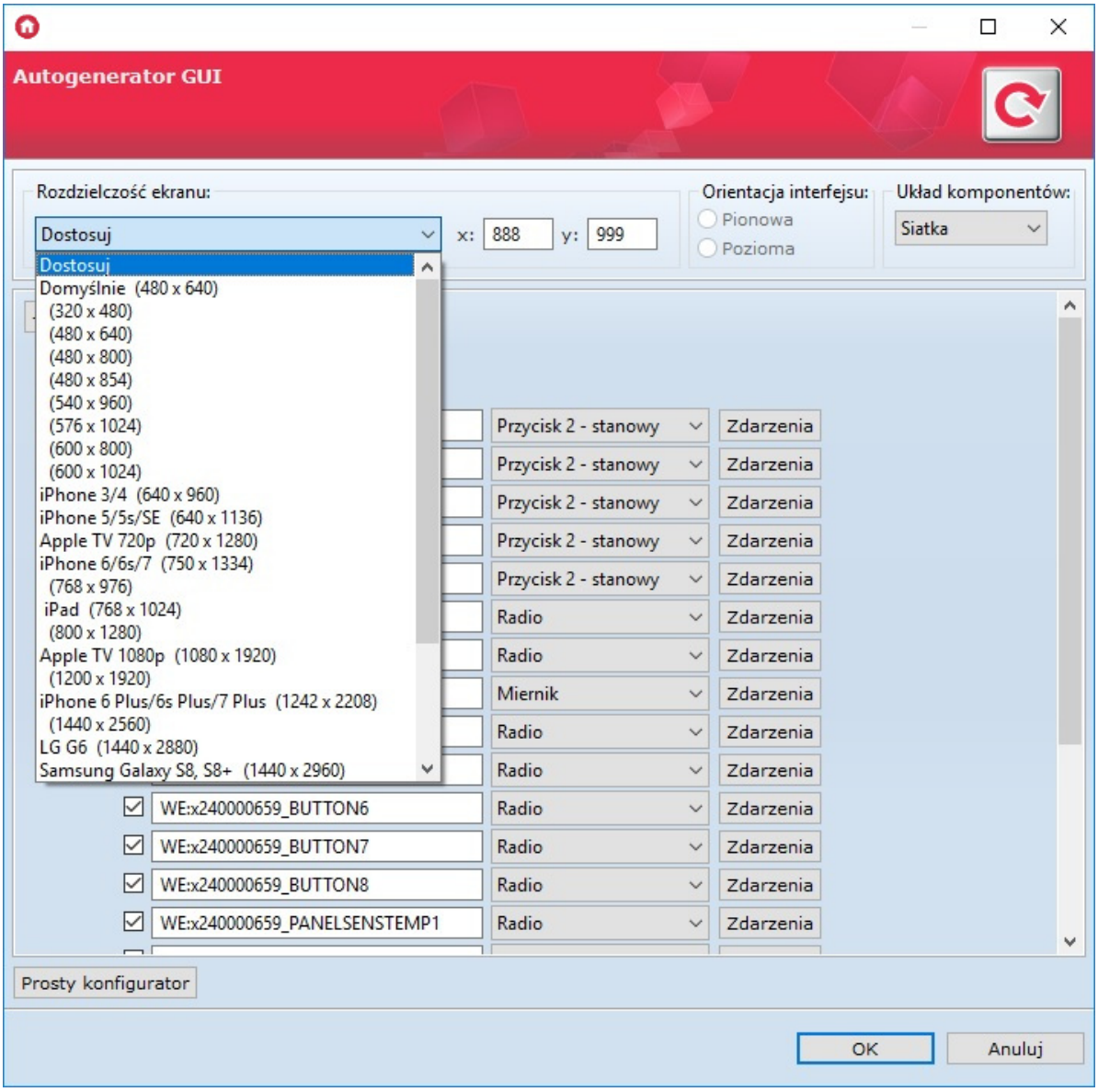


Podobnie jak w przypadku konfiguratora prostego - użytkownik ma możliwość dokonywania zmian w wygenerowanych stronach. Dwukrotne kliknięcie na ikonie strony otworzy arkusz edycji, zawierający dwie zakładki `Design` oraz `Symulator`.

5.2. Tworzenie interfejsu z własną rozdzielczością

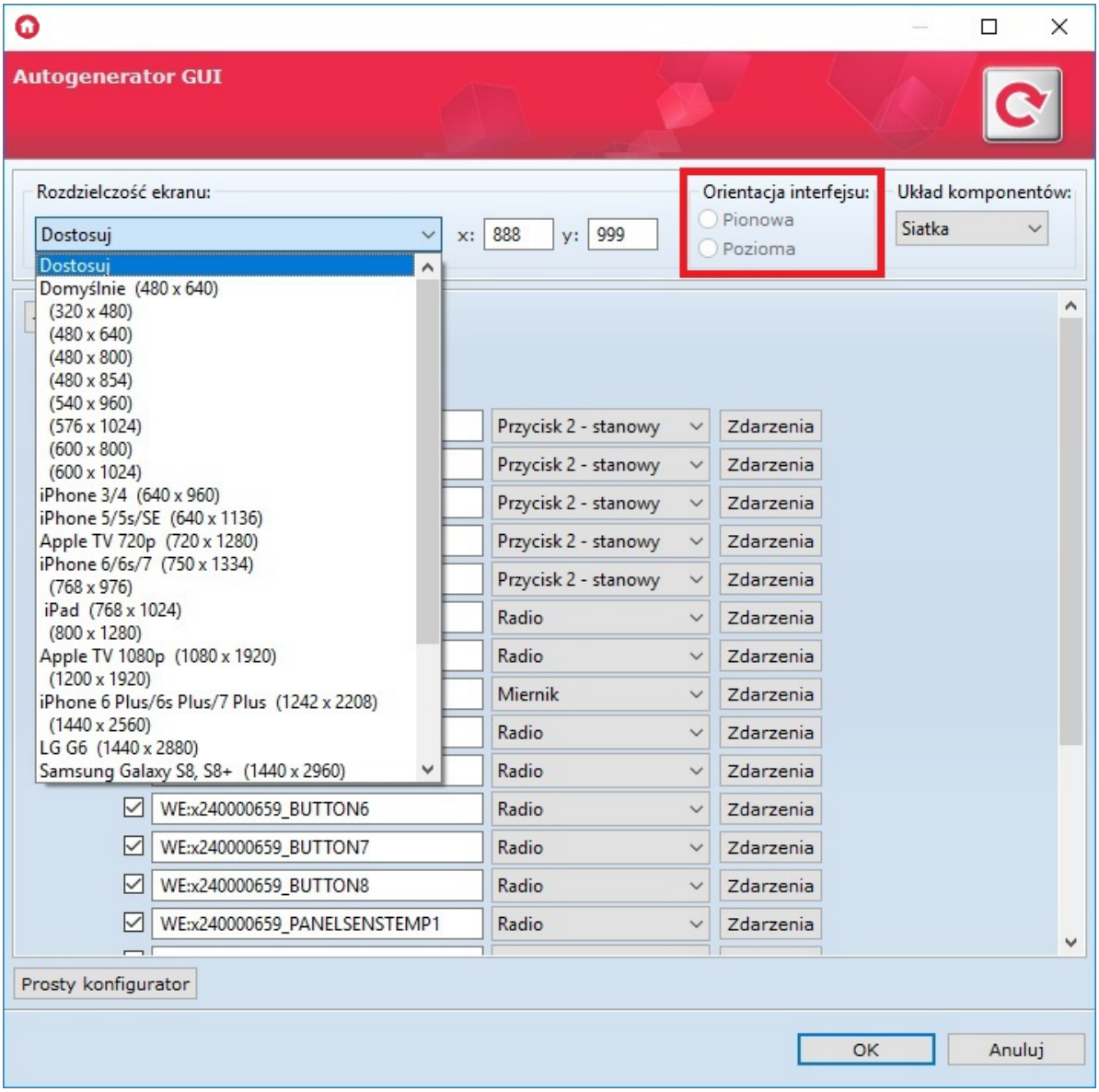
W przypadku konfiguratora zaawansowanego istnieje możliwość stworzenia interfejsu o własnej, wybranej rozdzielczości. W tym celu należy:

- Kliknąć w ikonę `Generuj GUI` w górnym oknie obiektów;
- Wybrać konfigurator zaawansowany;
- W oknie wyboru rozdzielczości zaznaczyć opcję *Dostosuj* oraz wprowadzić wymiary interfejsu;
- Wybrać pozostałe parametry interfejsu;
- Zaakceptować wprowadzone ustawienia.



5.3. Zmiana orientacji interfejsu z własną rozdzielczością

Korzystając z zaawansowanego konfiguratora zmiana orientacji interfejsu nie odbywa się w oknie `Autogeneratora GUI`.



Chcąc dokonać zmiany orientacji interfejsu z własną rozdzielczością, po jego utworzeniu należy kolejno:

- Kliknąć dwukrotnie na jego nazwę;
- Przejść do zakładki `Strony`;
- Usunąć wszystkie widoczne strony;
- Przejść do zakładki `Wyglad`;
- Wybrać orientację - poziomą lub pionową;
- Ponownie przejść do zakładki `Strony`;
- Dodać strony do interfejsu;
- Zaakceptować zmiany klikając OK;
- Wysłać interfejs do urządzenia mobilnego.

6. Konfiguracja wideodomofonu

> **Uwaga!** Wsparcie dla funkcjonalności Visual Builder zostało wstrzymane. Kreator interfejsów Home Manager zostanie usunięty w następnych wersjach Object Manager.

6.1. Podłączenie i konfiguracja wideodomofonu

Konfiguracja wideodomofonu z systemem Grenton jest możliwa dla urządzeń podłączonych do sieci wspólnej (*LAN*) lub korzystających ze zdalnego dostępu do danej sieci, umożliwiających użycie strumienia *rtsp* kamery IP wbudowanej w urządzenie. Do poprawnej konfiguracji wideodomofonu potrzebne są co najmniej dwa konta na serwerze *SIP*.

Przykładowa konfiguracja została wykonana na domofonie *Akuvox R26*.

> **Uwaga!** Panel Domofon dostępny jest dla Object Managera w wersji 1.2.0.180202 i wyższej.

A. Podłączenie wideodomofonu

Należy:

- Podłączyć wideodomofon do zasilania;
- Podpiąć wideodomofon przy pomocy kabla sieciowego RJ45 do routera.

B. Konfiguracja kamery

Panel wideodomofonu w aplikacji Grenton Home Manager wykorzystuje wizualizację kamery wbudowanej w urządzenie - chcąc mieć dostęp do obrazu kamery, należy wystawić odpowiedni port w ustawieniach sieci.

W celu dokonania konfiguracji portu, należy zalogować się do ustawień routera przy pomocy jego adresu IP w przeglądarce internetowej, dokonać odpowiednich zmian, po czym zapisać ustawienia:

- Należy wejść w ustawienia przekierowań [1];
- Odnaleźć ustawienia portów;
- Ustawić port wyzwalania i przekazywania na **554** [2] oraz protokół wyzwalania i przekazywania na **TCP**;
- Zapisać ustawienia;

> **Uwaga!** Należy pamiętać, że w celu umożliwienia zdalnego połączenia aplikacji, konieczne jest ustawienie portu **1234** w protokole **UDP**.

- Na koniec należy przejść do listy aktualnie podłączonych urządzeń do sieci i zapisać adres IP wideodomofonu – będzie on potrzebny przy konfiguracji serwera *SIP*.

C. Konfiguracja SIP:

- Do stworzenia konfiguracji wideodomofonu potrzeba co najmniej dwa konta *SIP*;
- Przy pomocy przeglądarki należy zalogować się do wideodomofonu [3];
- Konieczne jest odszukanie ustawień konta *SIP* [4];
- Następnie należy wybrać jedno z dostępnych kont (np. **Konto_1**) i ustawić jego stan na aktywowany (`enabled`);
- W kolejnym kroku należy ustawić numer/nazwę konta *SIP* oraz jego hasło;
- Później konieczne jest wprowadzenie ustawień serwera *SIP* (`Server IP`, `Port`, `Registration Period`) – ustawienia te powinny się pojawić przy zakładaniu kont;

- Następnie należy odnaleźć ustawienia kodeków wykorzystywanych w działaniu i aktywować kodeki typu *PCMU*;
- Na końcu konieczne jest odnalezienie ustawień Intercomu, gdzie należy skonfigurować numer/nazwę klienta, na który ma dzwonić wideodomofon (drugie założone konto *SIP*) i ustawić (jeśli istnieje taka możliwość) zachowanie urządzenia w momencie nieodebrania połączenia.

> **Uwaga!** Jeśli przy ustawianiu Intercomu, konieczny jest wybór jednego konta z kilku konfigurowanych, należy wybrać wcześniej wybrane - w przykładzie **Konto_1**!

6.2. Stworzenie i konfiguracja interfejsu aplikacji

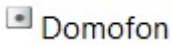
A. Dodawanie wideodomofonu do interfejsu aplikacji w programie Object Manager

W celu dodania wideodomofonu do interfejsu należy:

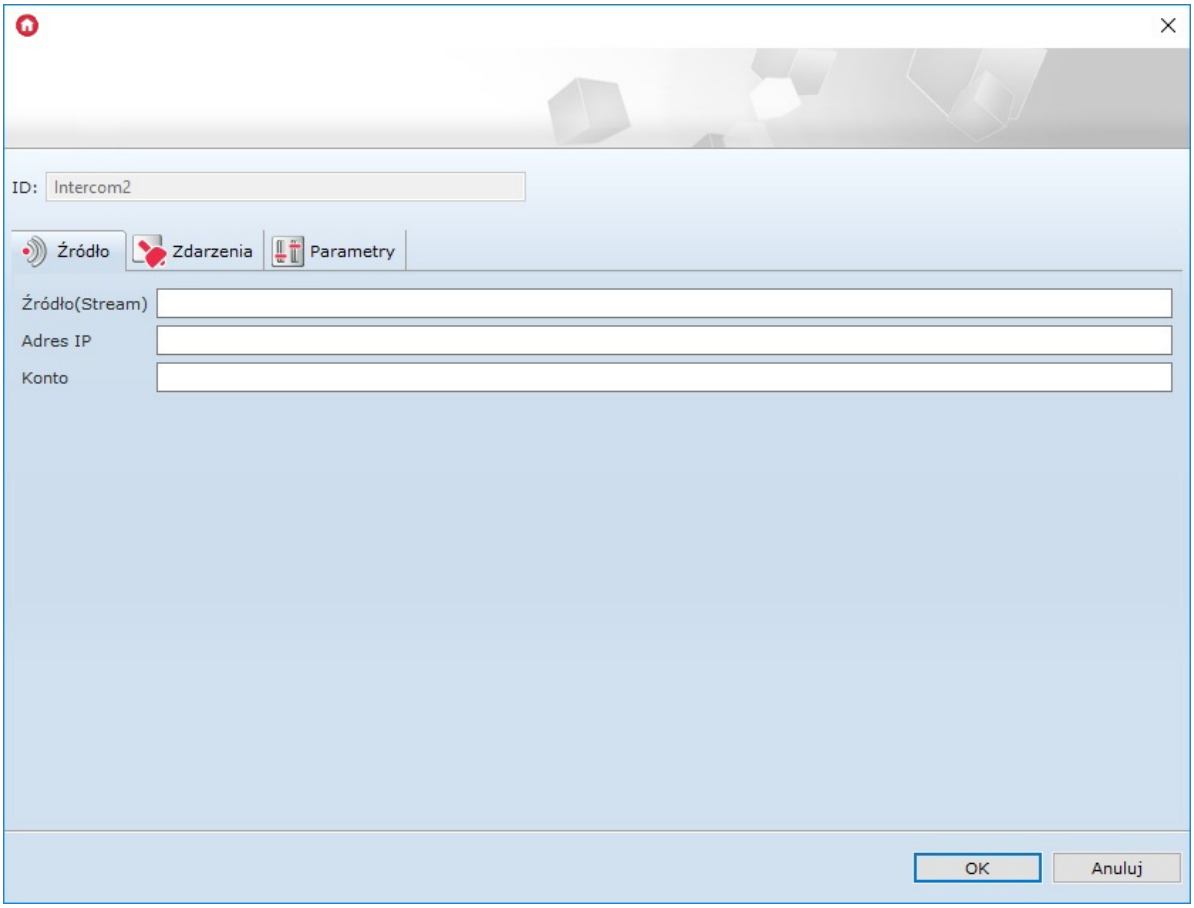
- Z menu głównego kliknąć przycisk *Dodaj interfejs*:



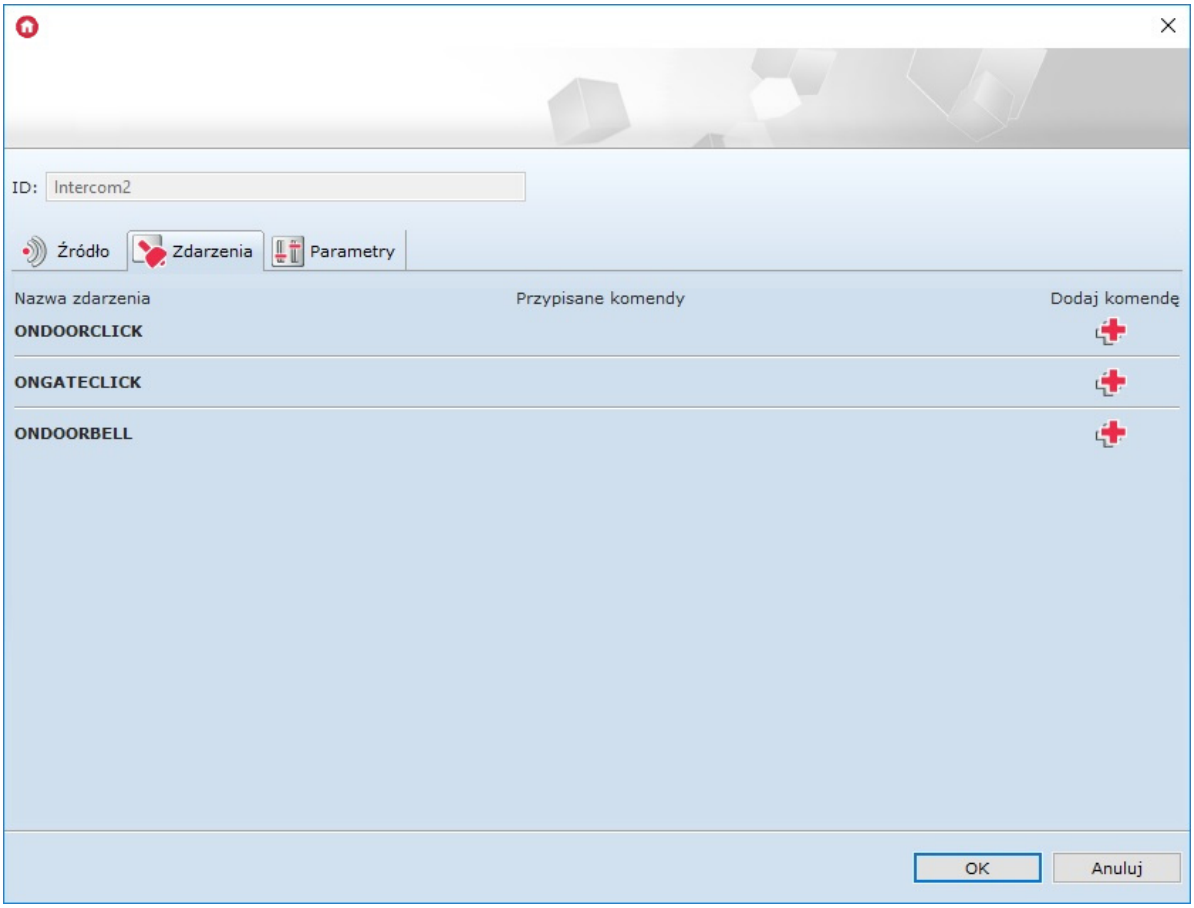
- Skonfigurować ustawienia interfejsu – wybrać: rozdzielczość, nazwę, skórkę, dodać przynajmniej jedną stronę;
- Do utworzonej strony - z palety komponentów - dodać przycisk *Domofon*:



- W oknie, które otworzy się po dodaniu przycisku, ustawić parametry wideodomofonu:
  - **Źródło** – strumień *rtsp* odnaleziony w ustawieniach wideodomofonu lub jego dokumentacji;
  - **Adres IP** – adres IP wideodomofonu (zapisany wcześniej przy dokonywaniu jego konfiguracji);
  - **Konto** – numer/nazwa konta *SIP* wprowadzonego jako pierwsze w ustawieniach wideodomofonu – konto, z którego będą wykonywane połączenia (wybrane w punkcie 3 rozdziału „*Podłączenie i konfiguracja wideodomofonu*"):



- Przejść do zakładki *Zdarzenia*:
    - Do zdarzenia `OnDoorClick` przypisać metodę, która ma być wywoływana po naciśnięciu przycisku otwierania furtki w panelu domofonu w aplikacji Home Manager;
    - Zdarzenie `OnGateClick` powiązać z metodą, która ma być wywoływana po naciśnięciu przycisku otwierania bramy wjazdowej w panelu domofonu w aplikacji Home Manager;
    - Do zdarzenia `OnDoorBell` przypisać metodę lub skrypt, które mają być wykonywane w momencie wykonania połączenia – w chwili naciśnięcia dzwonka na domofonie:



- Kliknąć *OK*;
- Wysłać interfejs na urządzenie mobilne - patrz pkt VIII.4.7..

B. Konfiguracja aplikacji Home Manager

W celu przeprowadzenia konfiguracji należy:

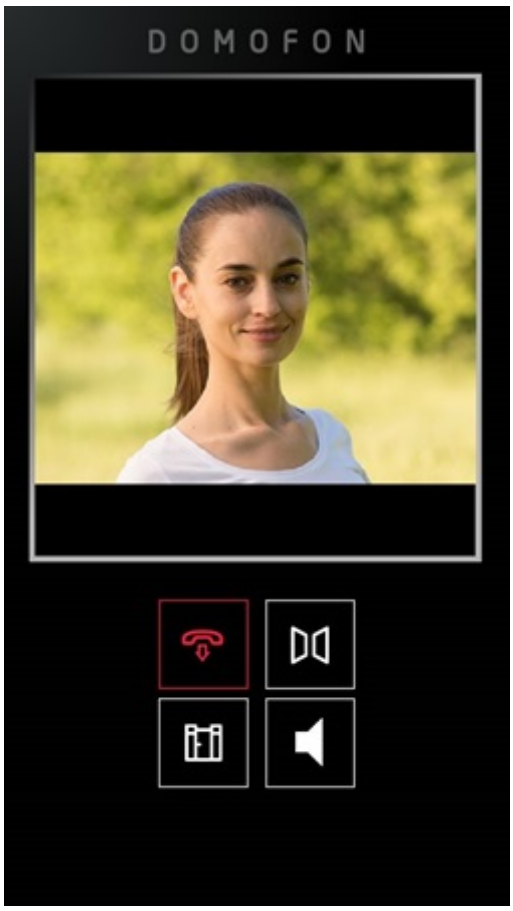
- Otworzyć aplikację Home Manager;
- Z menu głównego wybrać *Ustawienia* (piktogram koła zębatego);
- Z sekcji *Domofon* wybrać *Konfiguracja SIP* [5];
- W ustawieniach podać:
  - **Adres serwera** – adres IP serwera *SIP*, na którym zostały założone konta;
  - **Nazwę użytkownika** – numer/nazwę konta *SIP*, na które będą wykonywane połączenia – podanego w ustawieniach domofonu, jako konto docelowe do odbierania połączeń (wybrane w punkcie 3 rozdziału „Podłączenie i konfiguracja wideodomofonu”);
  - **Hasło** – hasło do powyższego konta *SIP*, na które będą wykonywane połączenia z domofonu:



- Zatwierdzić wprowadzone ustawienia przyciskiem *Zapisz*;
- Poprawnie przeprowadzona konfiguracja spowoduje, że na ekranie urządzenia mobilnego - w jego pasku powiadomień - pojawi się informacja o połączeniu z serwerem *SIP*;
- Wyjść z ustawień aplikacji.

6.3. Wykonanie połączenia z domofonu

1. Na domofonie nacisnąć przycisk dzwonienia.
2. Niezależnie od tego czy aplikacja Home Manager w urządzeniu mobilnym jest otwarta, zostanie nawiązane połączenie – na ekranie pojawi się panel wideodomofonu.
3. Przycisk po lewej stronie u góry służy do odbierania połączenia – dopóki nie zostanie użyty - rozmówca nic nie usłyszy, a domofon ciągle będzie dzwonił.
4. Z pozycji panelu wideodomofonu można wyzwolić zdarzenie `OnDoorClick` oraz `OnGateClick`, które zadziałają w zależności od ustawienia dokonanego w programie Object Manager.
5. W panelu domofonu znajduje się również przycisk, służący do załączania/wyłączania trybu głośnomówiącego.



7. Obsługa obrazu z kamer

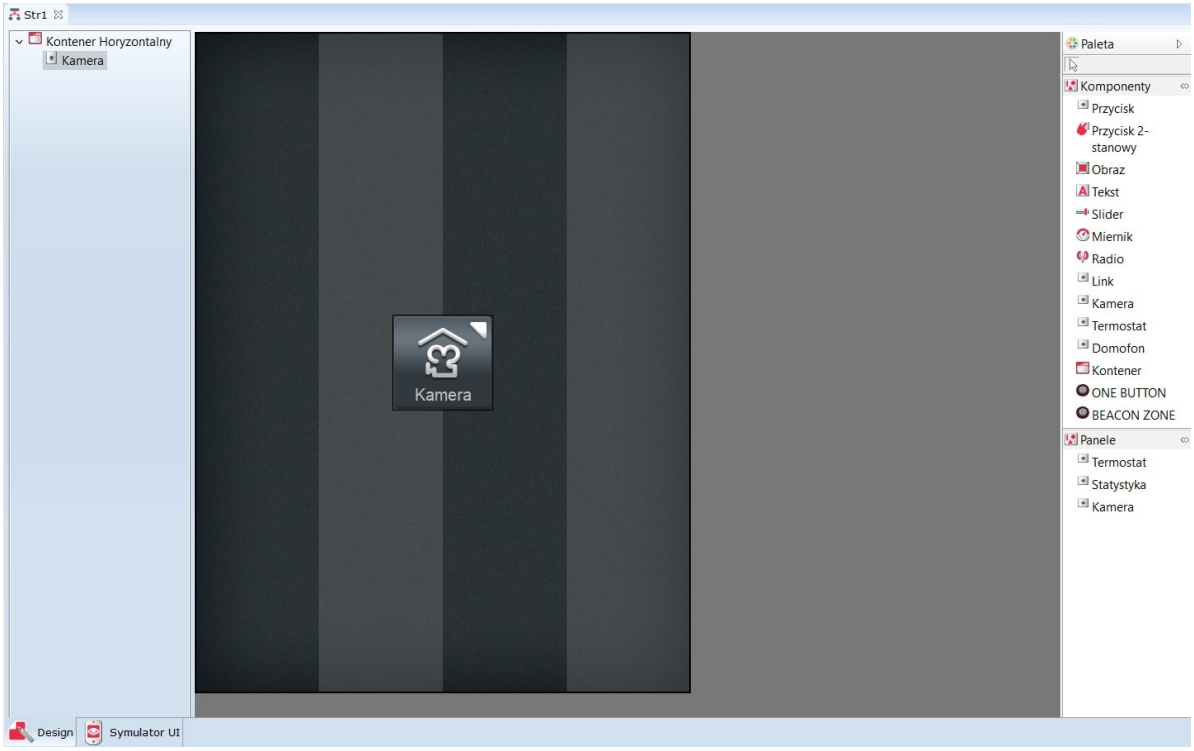
> **Uwaga!** Wsparcie dla funkcjonalności Visual Builder zostało wstrzymane. Kreator interfejsów Home Manager zostanie usunięty w następnych wersjach Object Manager.

Aplikacja Home Manager umożliwia dokonywanie podglądu obrazu z kamer IP w dowolnym interfejsie. Nie ma ograniczeń w liczbie obsługiwanych kamer, jednakże obraz z nich będzie wyświetlany pojedynczo.

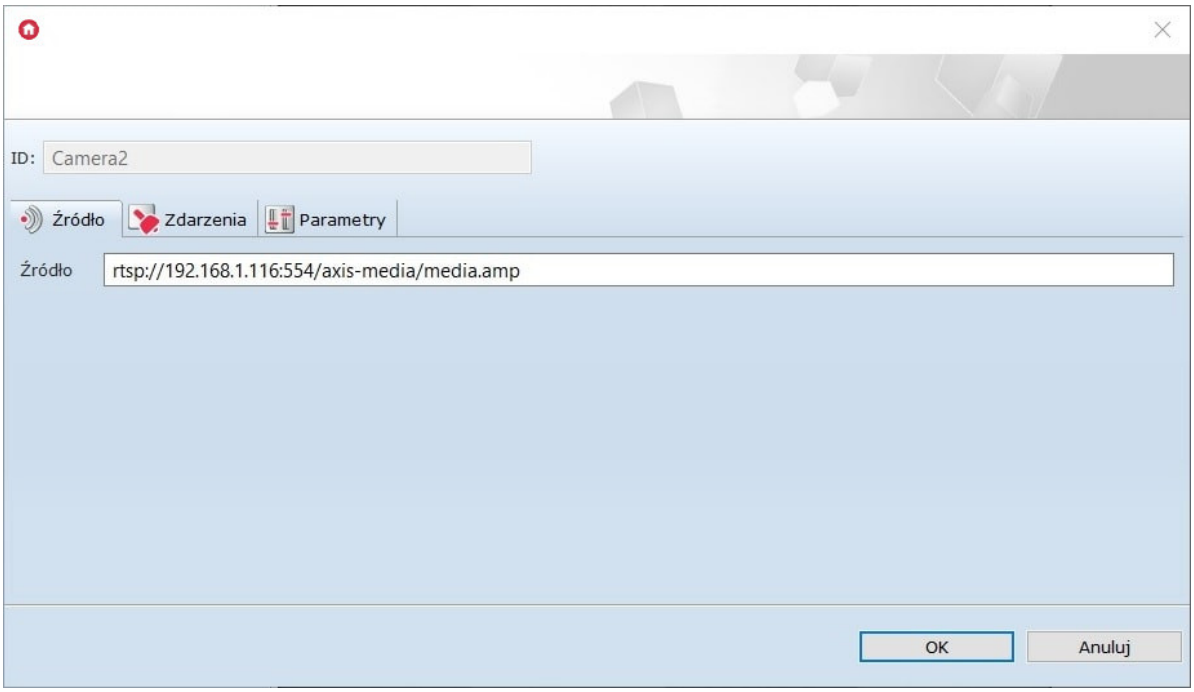
> **Uwaga!** Aplikacja Home Manager poprawnie wyświetla obrazy z kamer obsługujących protokół RTSP oraz kodek H.264 w strumieniu MPEG.

A. Dodawanie komponentu *Kamera*

Chcąc dodać do interfejsu obraz z kamery, należy przeciągnąć do obszaru roboczego komponent *Kamera* dostępny na liście obiektów:



Następnie - jako źródło dla dodanego obiektu, konieczne jest wpisanie adresu kamery, której obraz ma być wyświetlany. Dodawana kamera musi zostać uprzednio skonfigurowana w taki sposób, aby możliwe było otworzenie podglądu z niej za pośrednictwem protokołu RTSP.

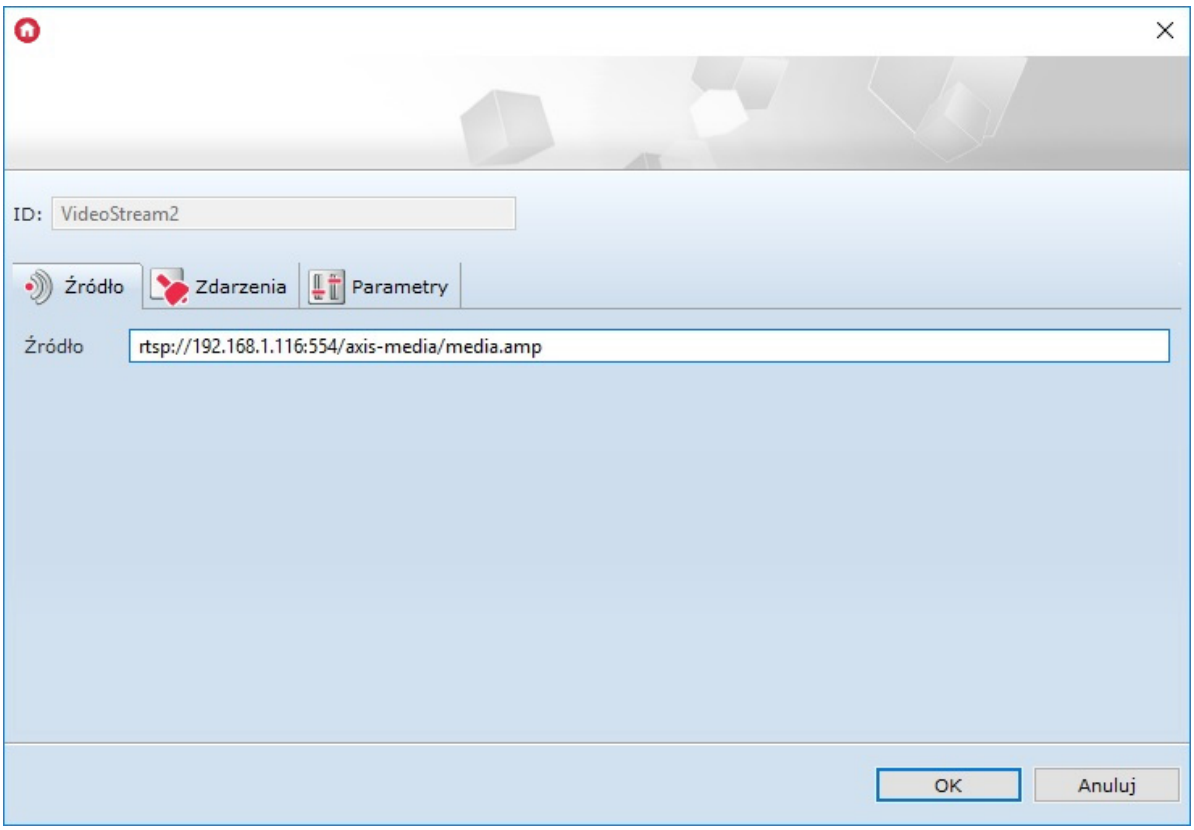


Po wysłaniu utworzonego interfejsu, obraz z kamery zostanie wyświetlony na ekranie urządzenia mobilnego po naciśnięciu wstawionego obiektu.

B. Dodanie panelu *Kamera*

Istnieje możliwość dodania do interfejsu obrazu z kamery przy pomocy panelu *Kamera*. W tym celu należy przeciągnąć go na pustą stronę interfejsu.

Następnie - jako źródło dla dodanego obiektu, konieczne jest wpisanie adresu kamery, której obraz ma być wyświetlany. Dodawana kamera musi zostać uprzednio skonfigurowana w taki sposób, aby możliwe było otworzenie podglądu z niej za pośrednictwem protokołu RTSP.



Po wysłaniu utworzonego interfejsu, obraz z kamery zostanie wyświetlony na ekranie urządzenia mobilnego po naciśnięciu strony z dodanym panelem *Kamera*.

8. Zdalny dostęp aplikacji mobilnej do systemu

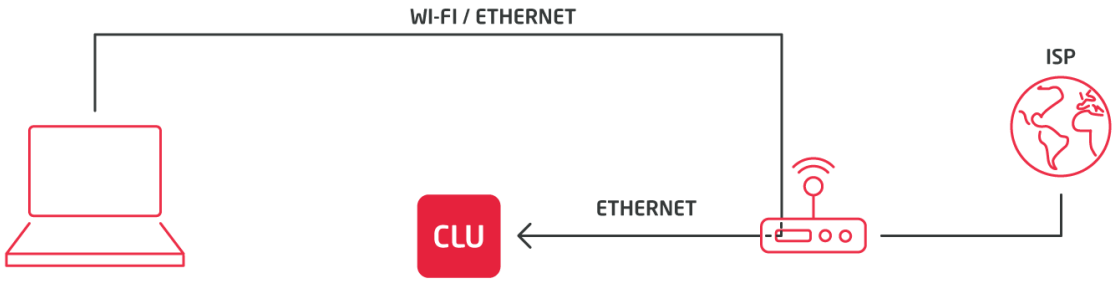
System Grenton daje możliwość swobodnego sterowania domem z każdego miejsca na świecie. Siedząc w pracy, czy będąc w podróży służbowej, możemy bez problemu kontrolować stan naszej inwestycji i zarządzać jego funkcjami w bardzo prosty sposób.

By zdalny dostęp do systemu Grenton był możliwy, powinien spełniać następujące wymagania:

- system Grenton musi być w pełni skonfigurowany;
- utworzone interfejsy aplikacji mobilnej muszą być przesłane do urządzeń mobilnych, z których realizowany ma być zdalny dostęp;
- dostawca usług internetowych musi zapewnić dostęp do stałego, zewnętrznego adresu IP;
- router/punkt dostępu musi mieć możliwość przekierowania portów.

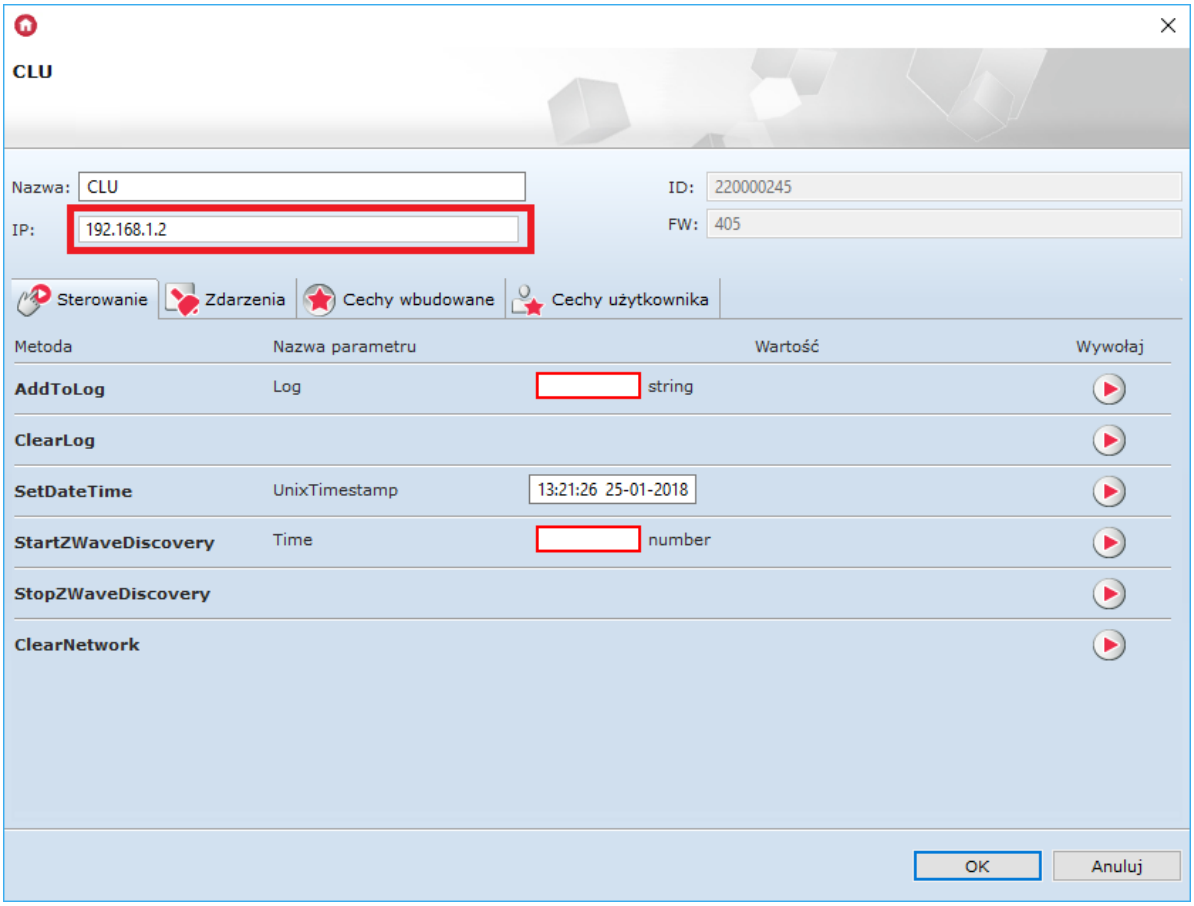
8.1. Konfiguracja systemu

Instrukcja została przygotowana dla układu, w którym jednostka centralna połączona jest do routera widocznego przez zewnętrzny, stały adres IP.



Przed przystąpieniem do konfiguracji zdalnego dostępu należy:

- upewnić się, czy jednostka centralna została połączona do routera sieci lokalnej oraz czy adres jednostki centralnej został nadany z puli adresów tego routera;
- sprawdzić adres jednostki centralnej nadanej przez router sieci lokalnej (w tym celu kliknąć dwukrotnie na ikonę jednostki centralnej);
- w nowo otwartym oknie odczytać informacje z zaznaczonego poniżej pola:



Dla analizowanego przypadku adres jednostki centralnej to: *192.168.1.2*. Adres ten będzie wykorzystywany do wykonania przekierowania portów.

8.2. Ustawienie przekierowania portów w routerze sieci lokalnej

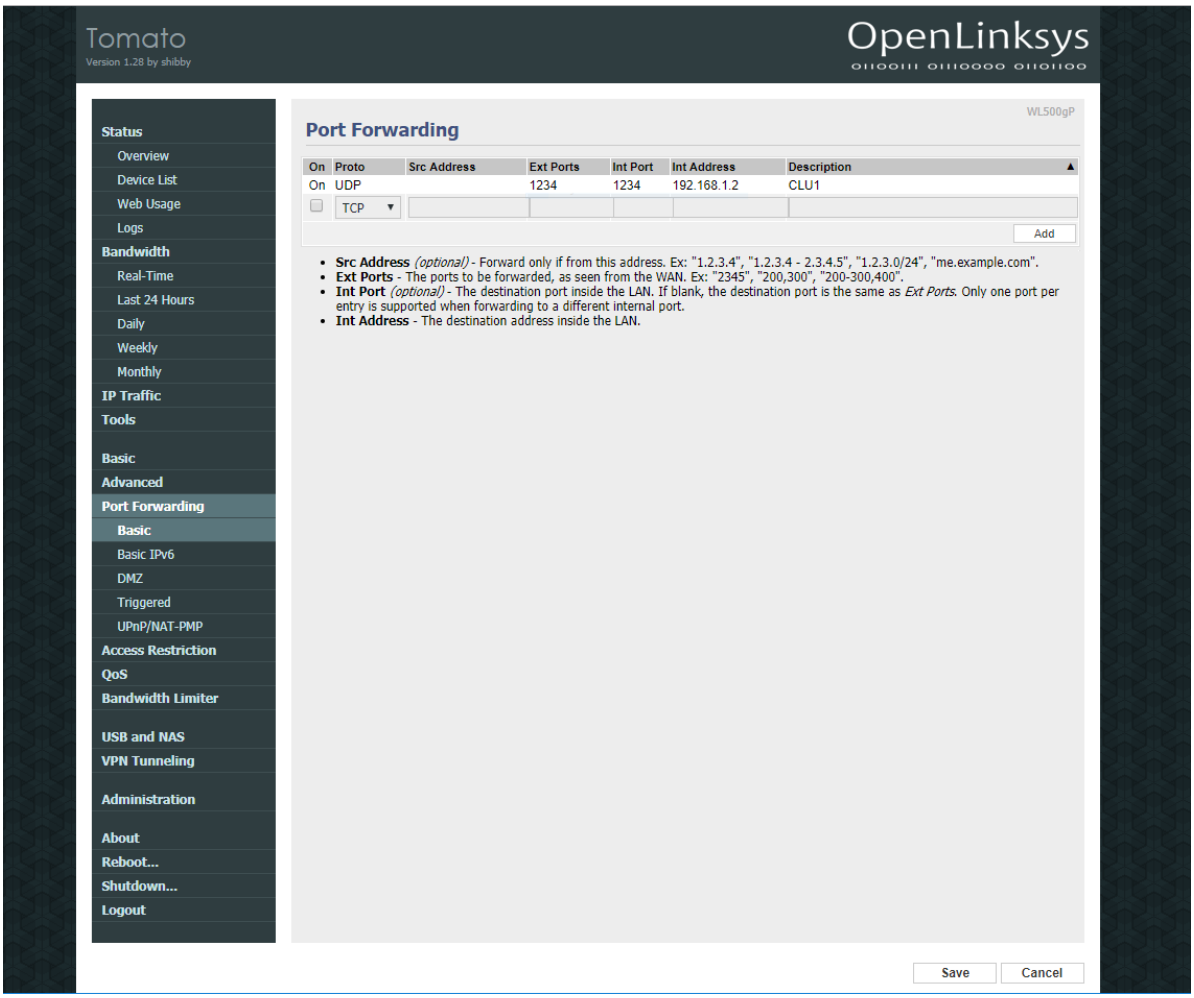
> **Uwaga!** Ustawienia przekierowania portów dla każdego routera mogą się różnić! Poniżej przedstawiona została ogólna procedura postępowania.

W celu ustawienia przekierowania portów konieczne jest:

- wejście do ustawień routera sieci lokalnej - by tego dokonać, wymagane jest podłączenie do sieci lokalnej, w której znajduje się jednostka centralna;
- otwarcie przeglądarki internetowej i wpisanie w polu adresu, adresu IP routera sieci lokalnej (w celu wejścia do jego ustawień) - domyślny adres znajduje się najczęściej na jego spodzie;
- zalogowanie się przy pomocy danych logowania - domyślny login i hasło znajdują się najczęściej w formie naklejki na spodzie routera sieci lokalnej (domyślne dane routera można również znaleźć w dedykowanych narzędziach internetowych);

> **Uwaga** Jeżeli wprowadzony adres IP lub dane logowania są nieprawidłowe, oznacza to, że zostały one zmienione przez administratora sieci. W celu uzyskania dostępu do ustawień routera należy się z nim skontaktować.

- odszukanie w ustawieniach routera pozycji dotyczącej przekierowania portów (*Port Forwarding* lub podobne);
- wykonanie przekierowania zewnętrznego portu 1234 na wewnętrzny port 1234 lokalnego adresu jednostki centralnej z wykorzystaniem protokołu UDP - przykładowa konfiguracja została zamieszczona poniżej:



- zapisanie ustawień routera - w niektórych przypadkach konieczne może być zrestartowanie urządzenia.

> **Uwaga!** Należy się upewnić, czy komunikacja z zewnątrz nie jest blokowana przez wewnętrzne ustawienia routera.

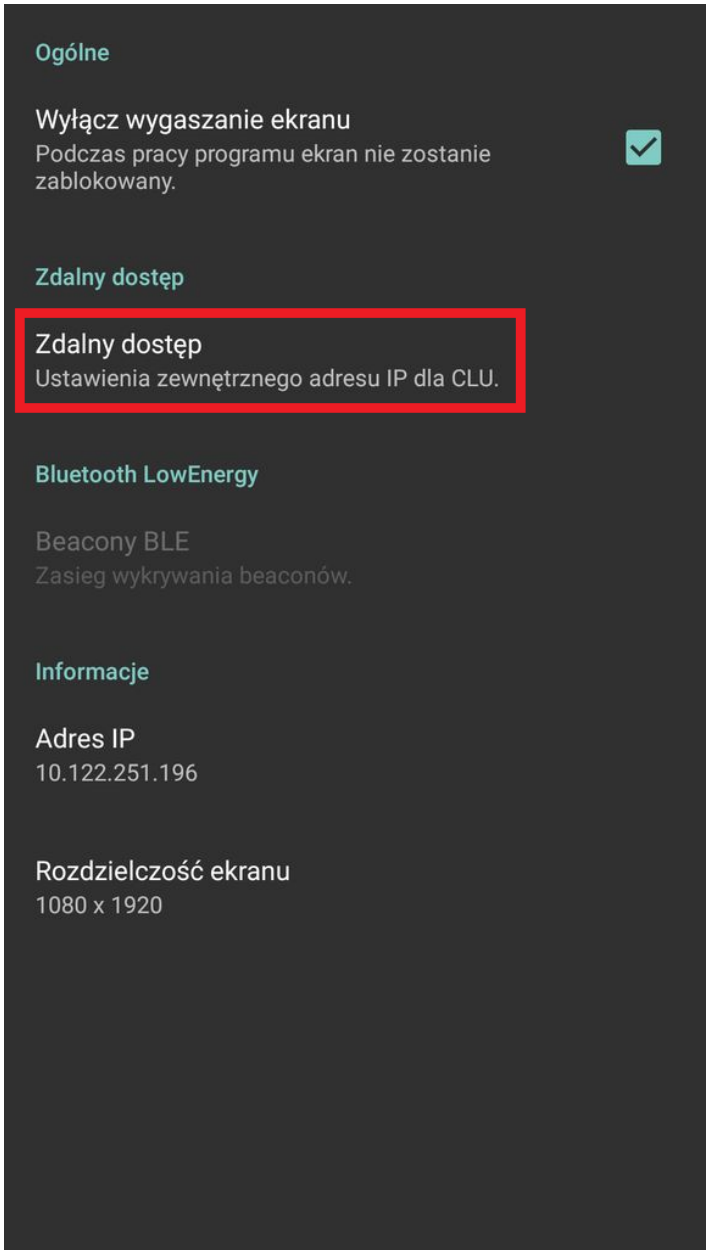
8.3. Konfiguracja aplikacji mobilnej Home Manager

Przy tworzeniu konfiguracji należy kolejno:

- uruchomić aplikację mobilną Home Manager;
- upewnić się, że do aplikacji mobilnej wgrany został interfejs, za pomocą którego realizowana będzie funkcjonalność zdalnego dostępu;
- przejść do ekranu głównego aplikacji mobilnej i wejść do *Ustawień* (klikając w ikonę koła zębatego znajdującą się w lewym dolnym rogu ekranu):



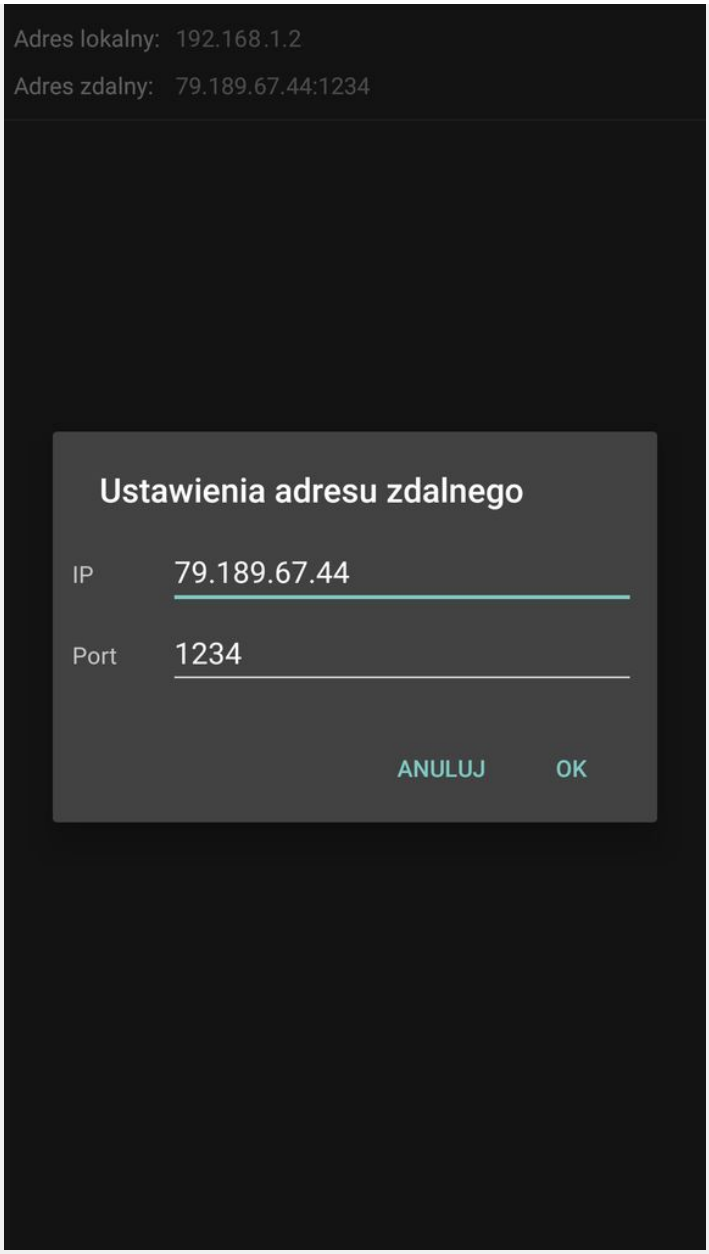
- w ustawieniach kliknąć w pozycję *Zdalny dostęp, Ustawienia zewnętrznego adresu IP dla CLU*:



- z listy dostępnych interfejsów wybrać ten, dla którego ma zostać skonfigurowany zdalny dostęp;
- następnie wyświetli się okno z aktualną konfiguracją sieciową systemu z informacjami o adresie:
    - lokalnym (lokalny adres IP jednostki centralnej);
    - zdalnym (zewnętrzny adres IP sieci, do której podłączona jest jednostka centralna wraz z numerem portu, który został jej przypisany):



Uwaga! Jeżeli podany adres zdalny różni się od rzeczywistego zewnętrznego adresu IP, należy dokonać zmiany klikając w okno adresu. W nowo otwartym oknie konieczne jest dokonanie zmian zgodnie z rzeczywistym adresem IP urządzenia. W celu zaakceptowania zmian należy nacisnąć *OK*.

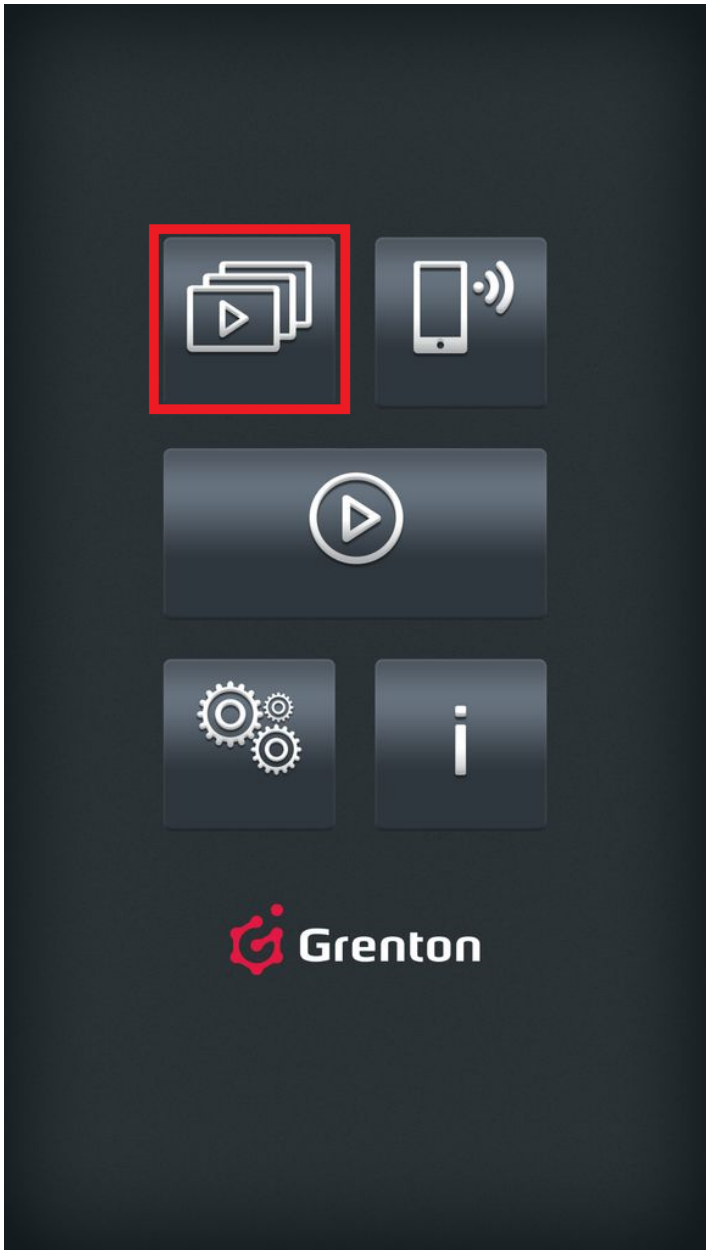


8.4. Uruchomienie zdalnego dostępu

Aplikacja mobilna Grenton Home Manager automatycznie przełącza się z komunikacji lokalnej na komunikację zdalną. Aby zdalny dostęp był możliwy, urządzenie mobilne musi spełniać następujące warunki:

- zdalny dostęp musi być poprawnie skonfigurowany;
- urządzenie musi być podłączone do sieci internetowej innej niż lokalna (inna niż ta, do której podłączony jest system) lub musi mieć włączone dane sieci komórkowej (*internet w telefonie*).

W celu uruchomienia zdalnej komunikacji z systemem należy otworzyć interfejs, dla którego wykonywana była konfiguracja zdalnego dostępu poprzez wybranie go z listy interfejsów:



Jeśli dany interfejs był ustawiony jako domyślny należy kliknąć przycisk:



W pierwszej kolejności aplikacja Home Manager podejmie próbę nawiązania połączenia poprzez sieć lokalną. W momencie wykrycia braku takiej możliwości nastąpi przełączenie na komunikacje zdalną.

IX. Obiekty CLU

1. Timer

Timery są wirtualnymi obiektami tworzonymi w ramach danego CLU. Object Manager umożliwia utworzenie maksymalnie 64 timerów. Timery mogą być wykorzystywane wszędzie tam, gdzie potrzebne jest wywołanie metody po określonym czasie lub też jej cykliczne wywoływanie. Sam timer jest również obiektem w OM i tak jak każdy inny obiekt posiada: swoje cechy, metody i zdarzenia oraz wartości początkowe. Timer może pracować w dwóch trybach:

- **Countdown**- po wystartowaniu, odlicza ustalony czas. Po zakończeniu odliczania uruchamiana jest metoda powiązana ze zdarzeniem `OnTimer`, a timer zatrzymuje się i nie odlicza, aż do następnego uruchomienia metodą `Start`.
- **Interval** - timer cykliczny - po starcie zaczyna odliczać ustawiony czas. Po jego upłynięciu timer wywołuje metodę powiązaną ze zdarzeniem `OnTimer`, a sam timer ponownie zaczyna odliczać zadany czas. Sytuacja powtarza się, aż do momentu zatrzymania metodą `Stop`.

A. Tworzenie timerów

W celu utworzenia timera w danym CLU należy zaznaczyć to, w którym ma się pojawić, a następnie wybrać z menu górnego ikonę `Dodaj obiekt CLU`.



Po kliknięciu ikony pojawia się okno wyboru z listą dostępnych obiektów, na której należy odszukać i wybrać obiekt `Timer`. Po zaznaczeniu, wciśnięciu `OK`, konieczne jest nadanie nazwy nowemu timerowi. Następnie należy ustawić czas timera [wartość podawana jest w ms] oraz wybrać tryb pracy [*Countdown* lub *Interval*]. Ustawiony czas będzie jednocześnie czasem w warunkach początkowych. Utworzony timer pojawi się na liście obiektów zaznaczonego CLU.
Utworzony timer również jest obiektem CLU i dlatego podobnie jak inne, fizyczne obiekty, jest obsługiwany przez konfigurator obiektów - patrz pkt VIII.4.1..

B. Parametry konfiguracyjne obiektu Timer

CECHY

Nazwa	Opis
`Time`	Zliczany czas (w ms)
`Mode`	Tryb pracy timera: 0 – zliczenie w dół (countdown), 1 – cykliczny (interval)
`State`	Aktualny stan pracy timera: 0 – zatrzymany (stopped), 1 – liczy (counting), 2 – pauza (paused)
`Value`	Czas jaki pozostał do wyzwolenia zdarzenia OnTimer (w ms)

METODY

Nazwa	Opis
`SetTime`	Ustawia czas timera (w ms)
`SetMode`	Ustawia tryb pracy
`Start`	Uruchamia timer
`Stop`	Zatrzymuje timer
`Pause`	Pauza timera

ZDARZENIA

Nazwa	Opis
`OnTimer`	Zdarzenie wywoływane przy zliczeniu timera
`OnStart`	Zdarzenie wywoływane przy uruchomieniu timera
`OnStop`	Zdarzenie wywoływane przy zatrzymaniu timera
`OnPause`	Zdarzenie wywoływane przy pauzie timera

2. Kalendarz

Kalendarze, podobnie jak timery, są wirtualnymi obiektami tworzonymi przez użytkownika w CLU. Możliwe jest utworzenie maksymalnie 64 kalendarzy na jednym CLU. Jeden kalendarz utworzony na CLU to jedna reguła wykonywana w określonym dniu i godzinie lub generowana cyklicznie w wybrane dni, miesiące czy godziny (z dokładnością co do jednej minuty). Reguły można tworzyć przy pomocy interfejsu graficznego lub też używając składni zgodnej z regułami CRON systemu LINUX.

A. Tworzenie kalendarza

Chcąc utworzyć kalendarz należy zaznaczyć CLU, w ramach którego ma się znaleźć, a następnie z menu górnego uruchomić `Dodaj obiekt CLU`.

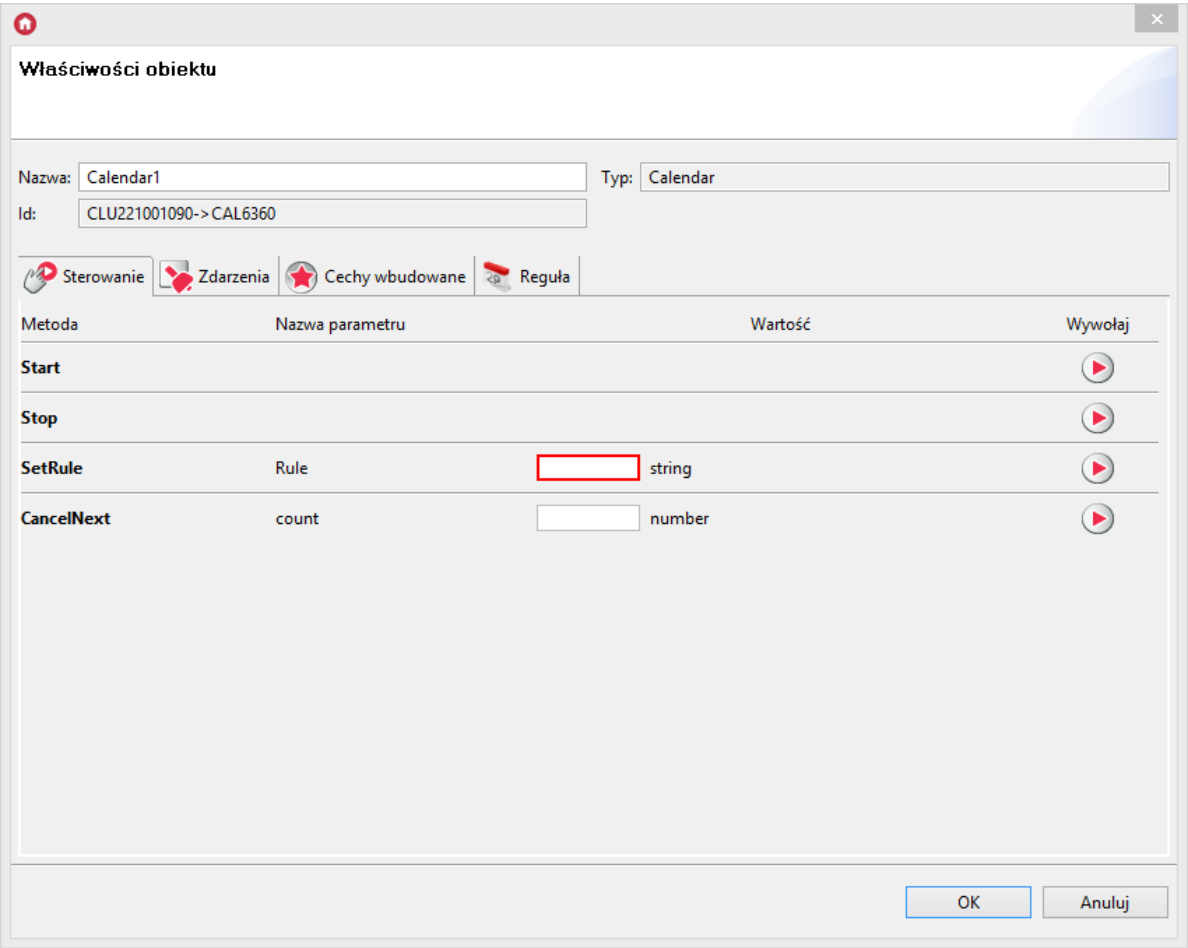


W otwartym oknie wybrać `Calendar`. Po wciśnięciu `OK`, wpisać nazwę dla tworzonego kalendarza. Object Manager wyświetli okno właściwości utworzonego obiektu.

> **Uwaga!** Kalendarz po utworzeniu i wysłaniu konfiguracji do CLU automatycznie staje się aktywny - aby zatrzymać pracę kalendarza należy wywołać metodę `STOP`.

B. Właściwości kalendarza

Okno właściwości kalendarza zawiera cztery zakładki:



- **Sterowanie** - zawiera metody kalendarza;
- **Zdarzenia** - zawiera zdarzenia kalendarza;
- **Cechy wbudowane** - zawiera listę cech kalendarza;
- **Reguła** - zawiera interfejs umożliwiający w prosty sposób formułowanie reguł.

C. Reguły kalendarza

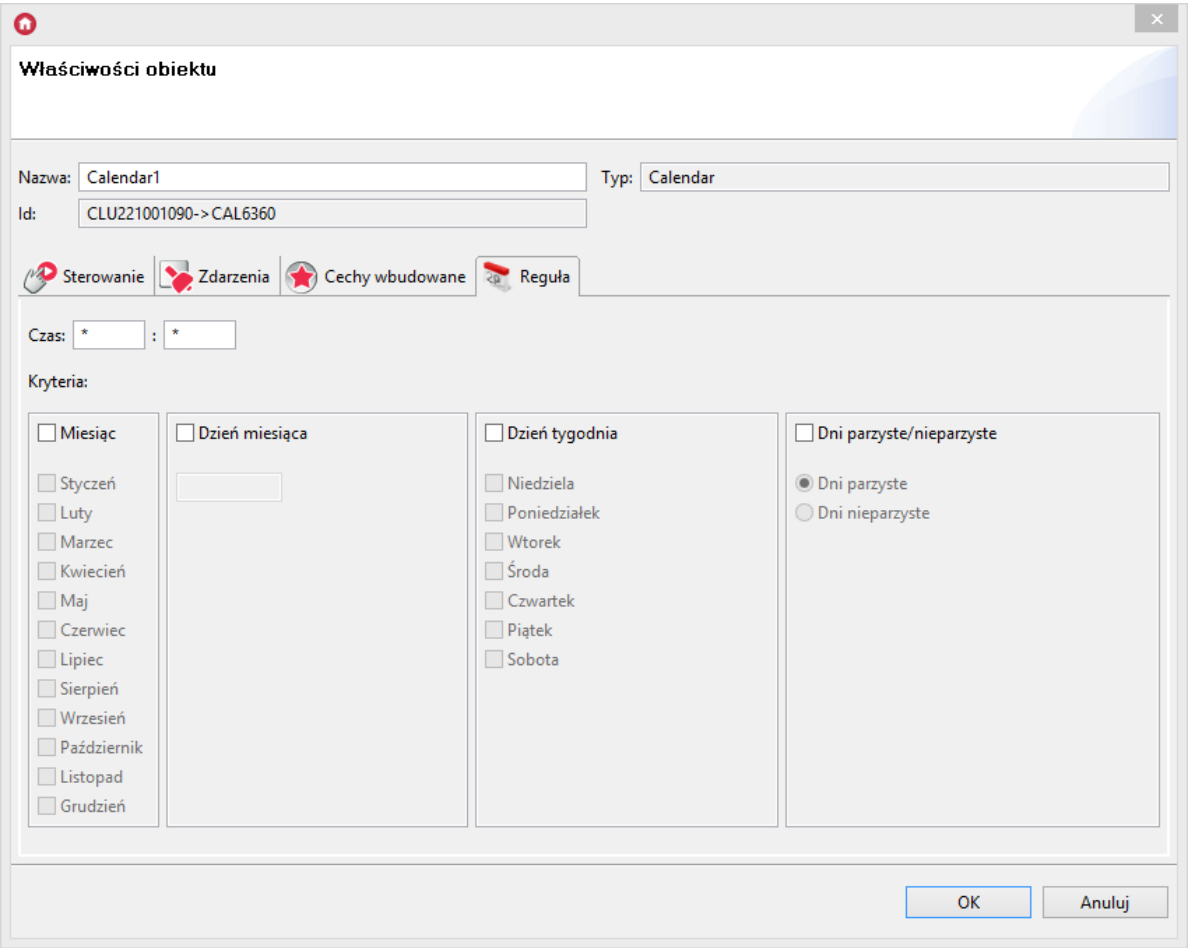
Wpisywanie reguł dla kalendarza może odbywać się na dwa sposoby:

- za pomocą interfejsu graficznego w zakładce `Reguła`;
- poprzez wpisanie reguły CRON z wykorzystaniem metody `SetRule` w zakładce sterowanie lub `Rule` jako cechy wbudowanej.

D. Tworzenie reguły kalendarza za pomocą interfejsu graficznego

W zakładce `Reguła` znajduje się graficzny interfejs, za pomocą którego użytkownik może w prosty sposób ustawić parametry reguły dla kalendarza.

> **Uwaga!** Po wpisaniu parametrów reguły za pomocą interfejsu graficznego, wartość `Rule` w `Cechach wbudowanych` wypełniana jest automatycznie zgodnie z zaznaczonymi kryteriami.



Znajdują się tam dwie sekcje, w których użytkownik wybiera parametry dla reguły:

- **Czas** - zawiera dwa pola: pierwsze, w którym wpisywana jest godzina (lub zakres godzin), druga oznacza minutę (lub zakres minut). Wartości w tych polach należy wpisywać zgodnie regułą CRON;
- **Kryteria** - zawiera pozostałe parametry dla reguły. Użytkownik dokonuje wyboru poprzez postawienie znacznika przy odpowiednim polu.

E. Tworzenie reguły kalendarza zgodnie z formatem CRON

Reguły kalendarza tworzy się poprzez wpisanie reguły w *Cechach wbudowanych* utworzonego kalendarza - w polu `Rule` lub poprzez metodę `SetRule`. Szczegółowe informacje na temat sposobu tworzenia reguł w formacie CRON można znaleźć w dokumentacji kalendarza CRON.

F. Parametry konfiguracyjne obiektu Kalendarz

CECHY

Nazwa	Opis
`Rule`	Reguła kalendarza w formacie CRON lub "ERROR" w przypadku wprowadzenia błędnej reguły
`SinceLastRun`	Czas (w minutach) od ostatniego spełnienia warunku reguły
`ToNextRun`	Czas (w minutach) do następnego wywołania akcji kalendarza
`State`	Stan działania kalendarza: 1 (kalendarz aktywny) lub 0 (nieaktywny)

METODY

Nazwa	Opis
`Start`	Przełączenie w stan aktywny (`State`=1)
`Stop`	Przełączenie w stan zatrzymania (`State`=0)
`SetRule`	Ustawienie reguły kalendarza
`CancelNext`	Anulowanie wywołania wskazanej liczby najbliższych akcji kalendarza

ZDARZENIA

Nazwa	Opis
`OnCalendar`	Zdarzenie generowane w momencie wywołania akcji kalendarza
`OnStart`	Zdarzenie generowane w momencie wznowienia pracy kalendarza
`OnStop`	Zdarzenie generowane w momencie zablokowania pracy kalendarza
`OnCancel`	Zdarzenie generowane w momencie anulowania najbliższych akcji kalendarza

3. Harmonogram

Harmonogram jest obiektem wirtualnym służącym do ustawiania wartości dowolnej cechy w rozkładzie tygodniowym. Wartości te ustawiane są przy pomocy interfejsu graficznego dla każdego dnia i każdej godziny z rozdzielczością 15-minutową, 30-minutową lub godzinną.
W jednym CLU, można utworzyć do 64 harmonogramów.

> **Uwaga!** Po utworzeniu harmonogramu (po wysłaniu nowej konfiguracji do CLU) staje się on automatycznie aktywny. Chcąc zatrzymać pracę harmonogramu, należy wywołać metodę `STOP`.

A. Tworzenie harmonogramu

W celu utworzenia harmonogramu należy zaznaczyć CLU, w ramach którego ma zostać umieszczony, a następnie z menu górnego uruchomić `Dodaj obiekt CLU`.

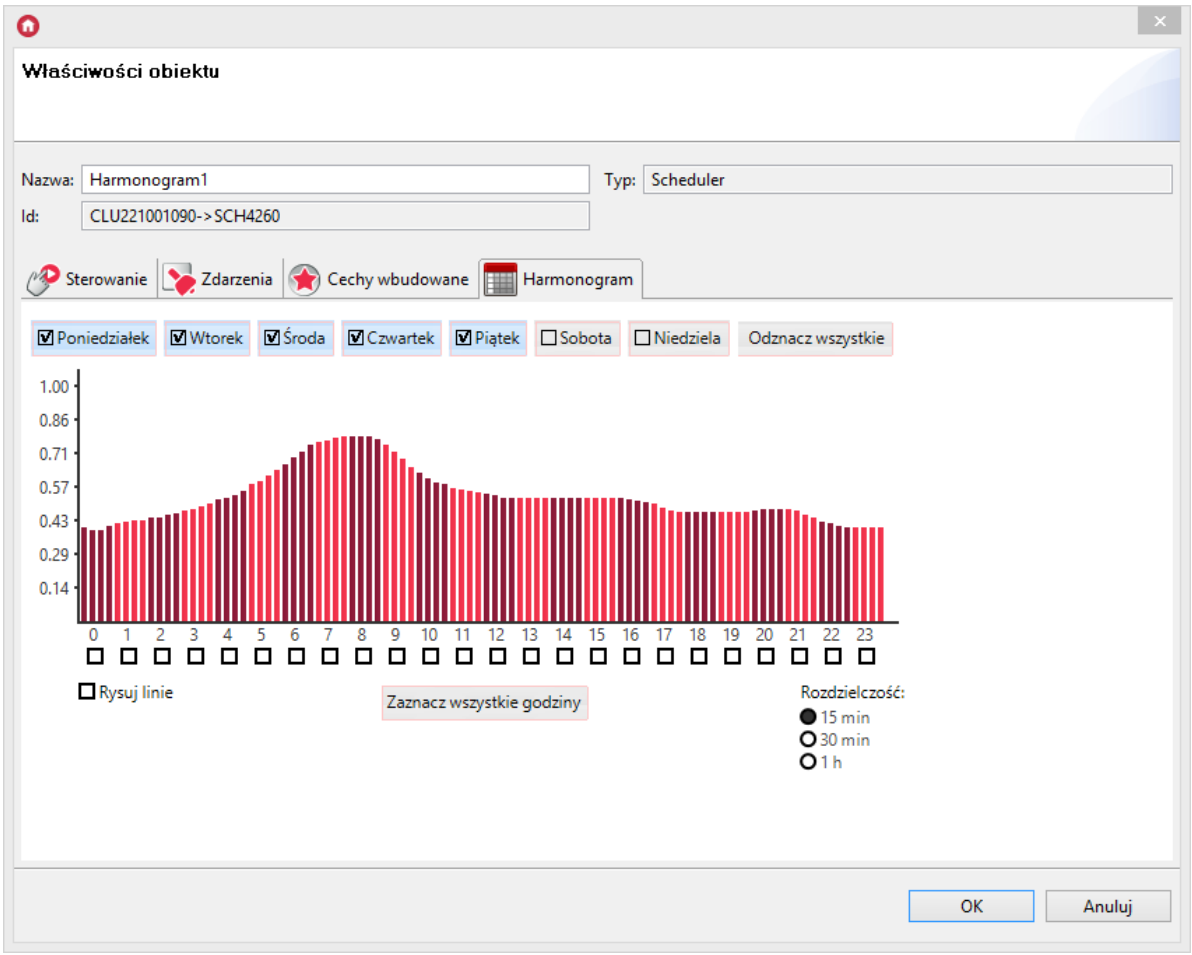


W otwartym oknie wyboru konieczne jest odszukanie i wybranie obiektu `Scheduler`. Po wpisaniu nazwy, na ekranie otworzy się okno właściwości harmonogramu. W oknie tym znajdują się cztery zakładki:

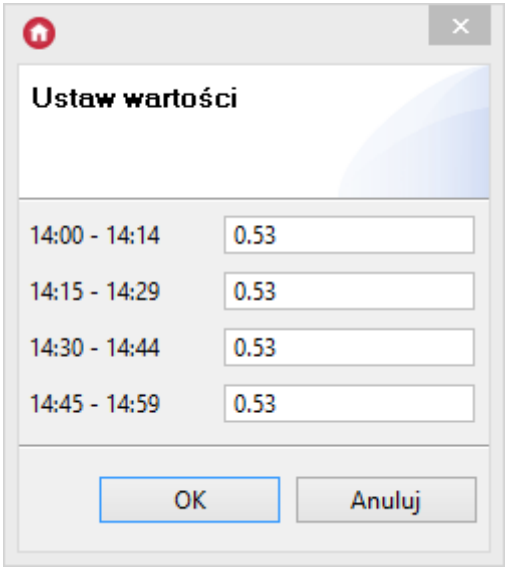
- **Sterowanie** - zawiera metody harmonogramu;
- **Zdarzenia** - zawiera zdarzenia harmonogramu;
- **Cechy wbudowane** - zawiera listę cech harmonogramu;
- **Harmonogram** - zawiera interfejs graficzny umożliwiający proste formułowanie wartości dla całego zakresu harmonogramu.

B. Formułowanie wartości dla harmonogramu

W zakładce *Harmonogram* (w oknie właściwości) znajduje się interfejs graficzny, dzięki któremu istnieje możliwość zdefiniowania wartości dla wybranego wyjścia.

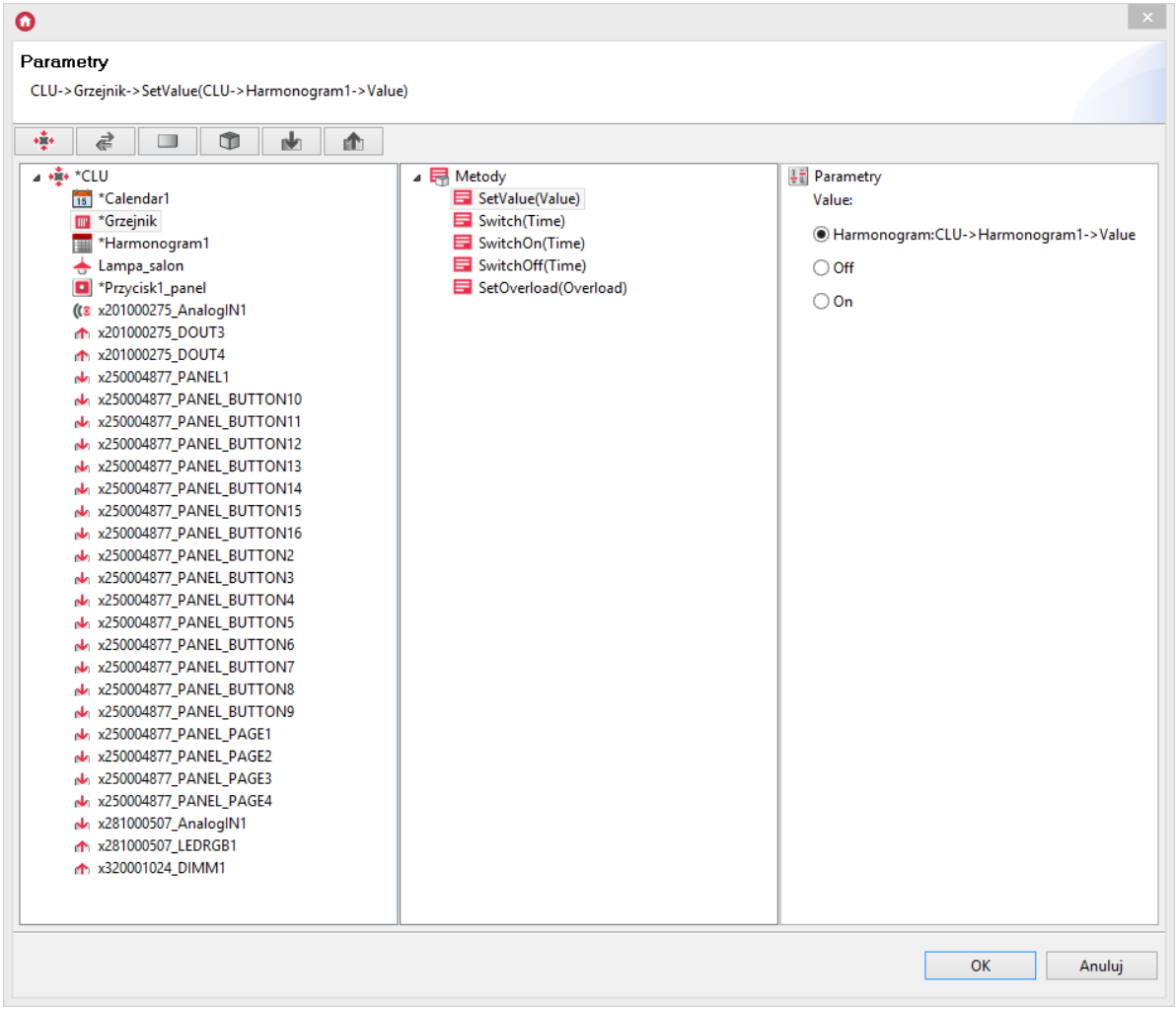


Harmonogram pozwala na ustawienie wartości dla 7 dni (w zakresie jednego tygodnia) z rozdzielczością 15-minutową. Można ustawić wartości dla każdego dnia z osobna lub kilku dni jednocześnie. Dzień, dla którego aktualnie wpisywane są wartości, wyróżniony jest czarnym znacznikiem z lewej strony nazwy. Przełączenie na inny dzień następuje po kliknięciu na jego nazwę.
W celu jednoczesnego wprowadzania wartości dla kilku dni, należy kliknąć znaczniki przy nazwach, dla których wartości będą ustawiane. Wartości można ustawiać bezpośrednio na wykresie posługując się myszką lub wpisywać ręcznie w oknie wartości, które otwiera się po kliknięciu na wybraną godzinę.



C. Ustawianie wartości wyjścia za pomocą harmonogramu

Zmiana wartości w ustawionym harmonogramie wywołuje zdarzenie `OnHarmonogram`. By wartości ustawione w harmonogramie zostały przepisane do wartości wybranego wyjścia, do zdarzenia `OnHarmonogram` trzeba dodać metodę `SetValue` dla wybranego wyjścia, a jako parametr tej metody należy wybrać `Harmonogram`.



Co 15 minut wartość (`Value`) dla tego wyjścia będzie ustawiana zgodnie z wartością zapisaną w harmonogramie.

> **Uwaga!** Należy pamiętać, aby zakres wartości ustawionych w harmonogramie odpowiadał zakresowi, w jakim wybrane wyjście może być sterowane. Zakres wartości harmonogramu można zmienić metodami `SetMax` i `SetMin` oraz poprzez zmianę cechy wbudowanych `Min` i `Max`.

D. Parametry konfiguracyjne obiektu Harmonogram

CECHY

Nazwa	Opis
`Data`	Ciąg znaków definiujący harmonogram zmian wartości
`State`	Stan działania harmonogramu: 1 (harmonogram aktywny) lub 0 (nieaktywny)
`Value`	Wartość wyjściowa, zmieniana co 15 minut zgodnie z rozkładem
`Min`	Minimalna wartość dla ustawiania zakresu wartości interfejsu graficznego
`Max`	Maksymalna wartość dla ustawiania zakresu wartości interfejsu graficznego

METODY

Nazwa	Opis
`Start`	Przełączenie w stan aktywny (`State`=1)
`Stop`	Przełączenie w stan zatrzymania (`State`=0)
`SetData`	Ustawienie tygodniowego harmonogramu

ZDARZENIA

Nazwa	Opis
`OnHarmonogram`	Zdarzenie wywoływane w momencie zmiany wartości cechy `Value`
`OnStart`	Zdarzenie wywoływane przy wznowieniu pracy
`OnStop`	Zdarzenie wywoływane przy zatrzymaniu pracy

4. Regulator PID

CLU umożliwia utworzenie do 64 regulatorów PID (proporcjonalno-całkująco-różniczkujących) służących do utrzymywania zadanej wartości wyjściowej na stałym poziomie w zależności od wartości wejściowej. Najpopularniejszym sposobem wykorzystania regulatora PID jest regulacja temperatury na podstawie informacji odczytywanej z czujnika temperatury.

> **Uwaga!** Regulator PID pracujący w trybie *AUTO* po rozpoczęciu pracy (po pierwszym uruchomieniu lub po resecie CLU) przeprowadza procedurę kalibracji obiektu, podczas której temperatura sterowanego obiektu może zostać podniesiona od kilku do kilkunastu procent powyżej temperatury zadanej. W związku z tym - nie zaleca się stosowania regulatorów PID dla obiektów o dużej bezwładności cieplnej, np. do ogrzewania lub kontrolowania temp. wody w akwariach.

A. Tworzenie regulatora

W celu utworzenia regulatora PID należy zaznaczyć CLU, w ramach którego ma się znaleźć, a następnie z menu górnego uruchomić `Dodaj obiekt CLU`.



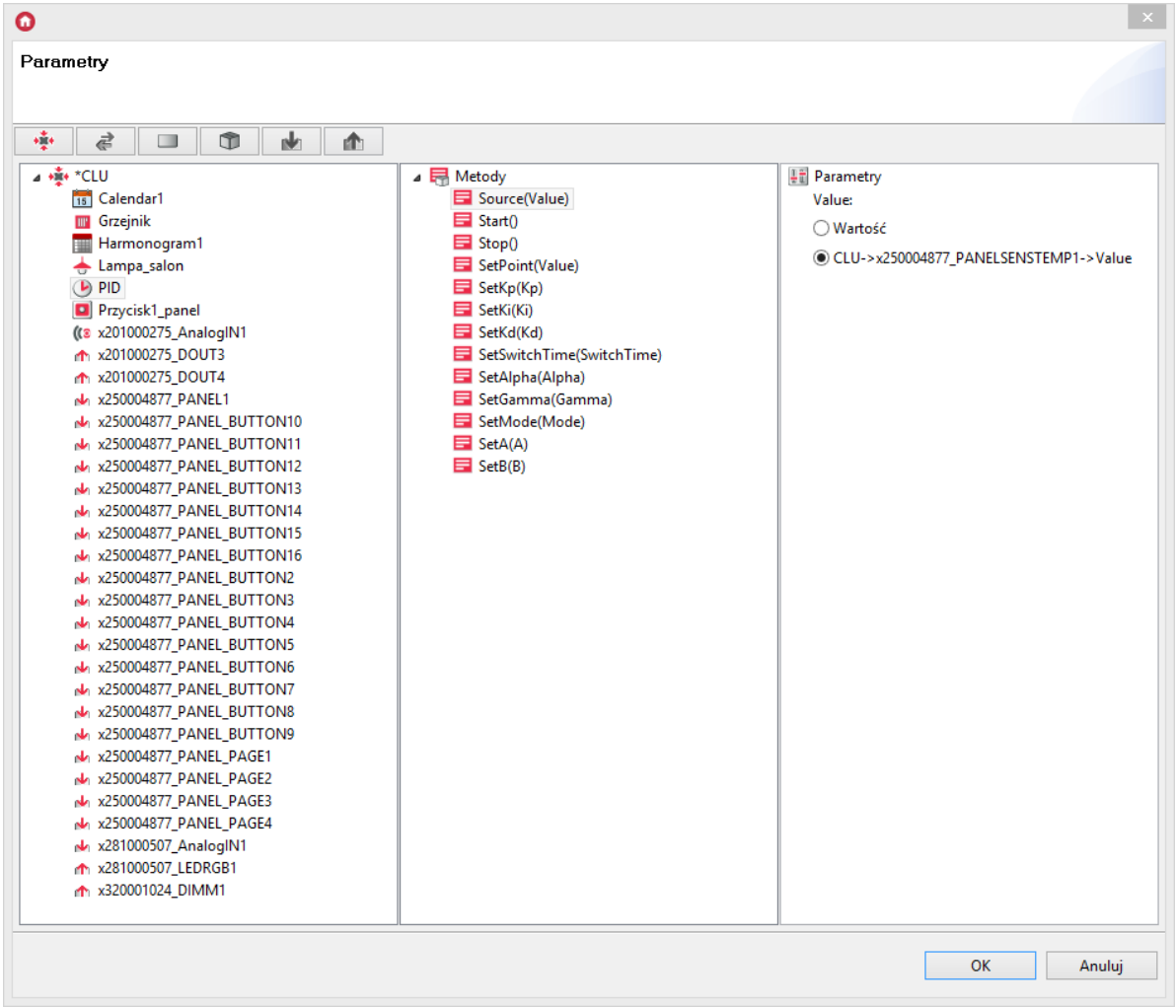
W otwartym oknie konieczne jest odnalezienie i zaznaczenie obiektu `PIDcontroler`, a następnie nadanie mu nazwy. Na ekranie pojawi się okno właściwości nowo utworzonego regulatora, w którym znajdują się trzy zakładki:

- **Sterowanie** – zawiera metody regulatora;
- **Zdarzenia** – zawiera zdarzenia regulatora;
- **Cechy wbudowane** – zawiera listę cech regulatora.

B. Sterowanie za pomocą regulatora

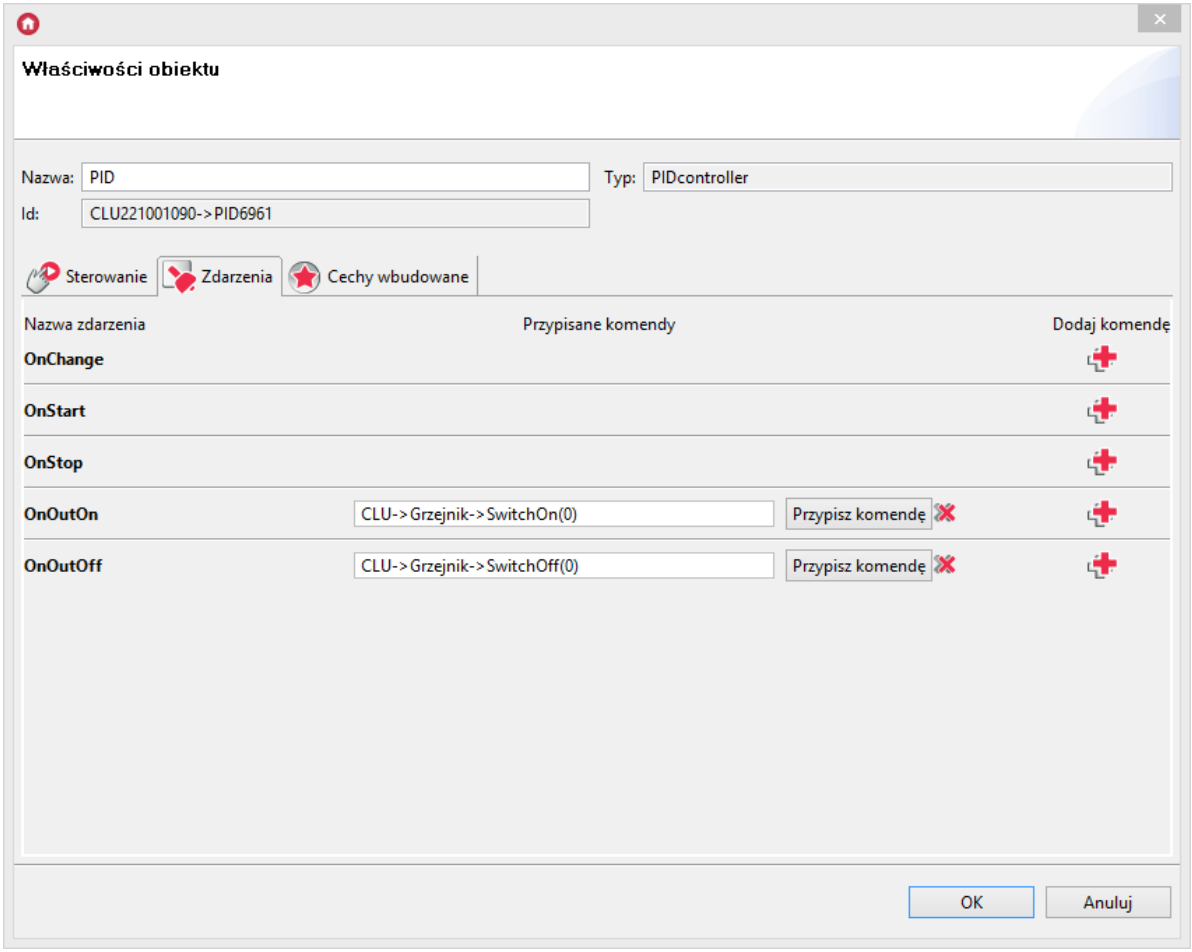
Sterowanie wartościami wyjściowymi z wykorzystaniem regulatora będzie możliwe, gdy zostanie poprawnie powiązany z obiektami wejściowym i wyjściowym. W tym celu kolejno:

- Do metody Source przypisać wartość źródłową, np. cechę `Value` czujnika temperatury (w czujniku temperatury do zdarzenia `OnChange` należy wybrać regulator PID, a do metody `Source`, jako parametr, przypisać wartość z czujnika temperatury).



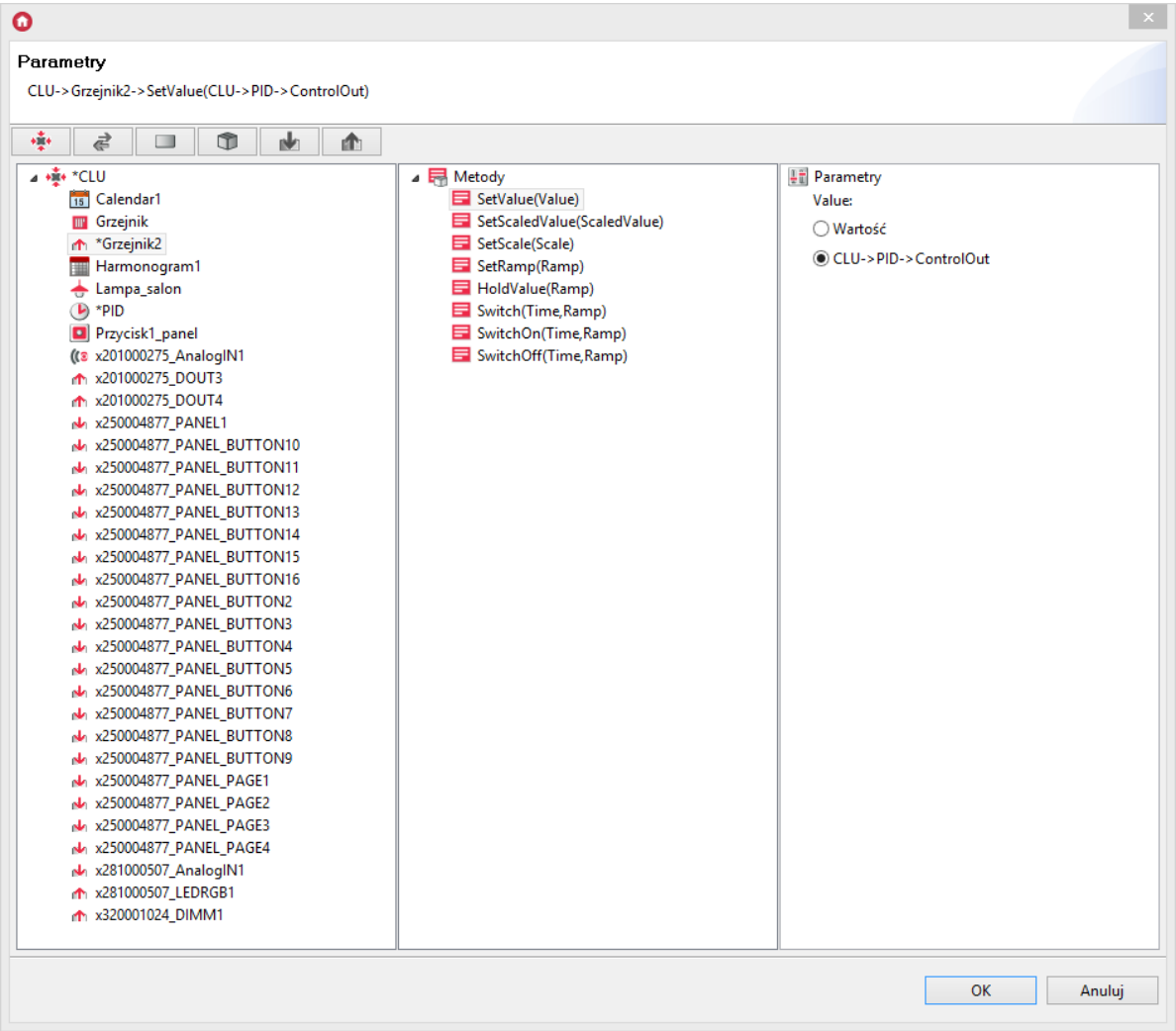
- Należy powiązać moduł wyjściowy z odpowiednimi zdarzeniami obiektu PID. By tego dokonać, do zdarzeń `OnOutOn` i `OnOutOff` przypisać metody sterujące dla obiektu.

W przypadku sterowania temperaturą, należy do zdarzenia `OnOutOn` w regulatorze przypisać metodę `SwitchOn` wyjścia, z którego sterowane jest źródło ciepła (do którego podpięty jest grzejnik, piec, zawór sterujący grzejnikiem), natomiast do zdarzenia `OnOutOff` należy przypisać metodę `SwitchOff` tego wyjścia.



Alternatywnie - jeśli interfejs modułu wyjściowego na to pozwala, można wykorzystać jedynie zdarzenie `OnChange`.

By to zrobić, należy do zdarzenia `OnChange` w regulatorze przypisać metodę `SetValue (Value)` wyjścia sterującego źródłem ciepła (dane wyjście musi posiadać taką metodę), a następnie jako parametr wskazać cechę `ControlOut` regulatora PID.



W takiej konfiguracji, gdy wartość wyjścia sterującego w regulatorze ulegnie zmianie, zostanie ona przepisana do wartości wyjścia.

C. Tryby pracy

Regulator ma dwa możliwe tryby pracy:

- **Sterowanie automatyczne (`Auto`)** Sterowanie w tym trybie oparte jest na algorytmie automatycznego strojenia, w którym wszystkie istotne parametry dostrajane są automatycznie na podstawie odczytywanych danych.
- **Sterowanie ręczne (`Normal`)** Użytkownik w tym trybie ma możliwość ustawienia wszystkich istotnych parametrów wykorzystywanych do sterowania w regulatorze PID z ręcznym doborem nastaw (parametry Kp, Ki, Kd).
  Tryb `Normal` przeznaczony jest dla zaawansowanych użytkowników znających zasady strojenia regulatorów PID.

Chcąc ustawić regulator w danym trybie pracy, zmień wartość cechy `Mode` na:

- `Normal` - dla trybu ręcznego;
- `Auto` - dla trybu automatycznego.

W zależności od wybranego trybu pracy zmienia się możliwość ustawiania wartości poszczególnych cech - np. parametry A i B wykorzystywane są tylko dla algorytmu `Auto`, a parametry Kp, Ki oraz Kd - tylko dla trybu `Normal`.

> **Uwaga!** Parametry A i B nie mogą być zmieniane w trakcie trwania sterowania, gdyż są one na bieżąco aktualizowane przez algorytm.

D. Zasada działania regulatora PID

Regulator steruje cechą `ControlOut` poprzez ustawianie jej wartości na 0 lub 1 z częstotliwością ustawiania cechy `SwitchTime`, poprzez zmianę wypełnienia impulsu. Przed rozpoczęciem sterowania, regulator najpierw przeprowadza procedurę szacowania bezwładności obiektu sterowanego i na tej podstawie wyznacza dopuszczalne granice wartości `SwitchTime`. Po zakończeniu tego etapu, wartość cechy `SwitchTime` ustawiana jest automatycznie w połowie wyznaczonego zakresu.

> **Uwaga!** W przypadku, gdy regulacja prowadzona jest w sposób automatyczny, ręczna zmiana wartości `SwitchTime` nie jest możliwa.

E. Parametry konfiguracyjne obiektu Regulator PID

CECHY

Nazwa	Opis
`ControlOut`	Wartość wyjścia sterującego (binarna, przełączana w cyklu określonym przez `SwitchTime`)
`State`	Stan działania regulatora: 1 (harmonogram aktywny) lub 0 (nieaktywny)
`SetPoint`	Wejście regulatora - wartość docelowa
`Kp`	Wzmocnienie członu proporcjonalnego regulatora PID
`Ki`	Wzmocnienie członu całkującego regulatora PID
`Kd`	Wzmocnienie członu różniczkującego regulatora PID
`SwitchTime*`	Czas przełączania
`Alpha`	Parametr α w algorytmie Kaczmarza (zabezpieczenie przed zerowaniem mianownika)
`Gamma`	Parametr γ w algorytmie Kaczmarza (dynamika zmian oszacowania a i b)
`Mode`	Tryb pracy regulatora: 1 - „ręczny" PID lub 2 - automatyczny algorytm Kaczmarza
`A*`	Parametr a w algorytmie Kaczmarza
`B*`	Parametr b w algorytmie Kaczmarza

- Ustawianie tych parametrów nie jest możliwe we wszystkich stanach pracy regulatora.

METODY

Nazwa	Opis
`Source`	Podanie nowej wartości wejścia dla sterownika (pętla sprzężenia zwrotnego)
`Start`	Przełączenie w stan aktywny (`State`=1)
`Stop`	Przełączenie w stan zatrzymania (`State`=0)
`SetPoint`	Ustawienie docelowej wartości regulatora
`SetKp`	Ustawienie wartości wzmocnienia członu proporcjonalnego
`SetKi`	Ustawienie wartości wzmocnienia członu całkującego
`SetKd`	Ustawienie wartości wzmocnienia członu różniczkującego
`SetSwitchTime`	Ustawienie czasu przełączania
`SetAlpha`	Ustawienie parametru Alpha w algorytmie Kaczmarza, zabezpieczającego przed zerowaniem mianownika
`SetGamma`	Ustawienie parametru Gamma w algorytmie Kaczmarza
`SetMode`	Ustawienie trybu pracy regulatora – ręczny PID (Normal PID) lub automatyczny algorytm Kaczmarza (Auto-Kaczmarz)
`SetA`	Ustawienie parametru a w algorytmie Kaczmarza
`SetB`	Ustawienie parametru b w algorytmie Kaczmarza

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w momencie zmiany wartości cechy `ControlOut`
`OnStart`	Zdarzenie wywoływane w momencie wznowienia pracy
`OnStop`	Zdarzenie wywoływane przy zablokowaniu pracy
`OnOutOn`	Zdarzenie wywoływane w momencie przełączenia wartości cechy `ControlOut` na 1
`OnOutOff`	Zdarzenie wywoływane w momencie przełączenia wartości cechy `ControlOut` na 0

5. Termostat

Termostat to obiekt wirtualny, który służy do stworzenia konfiguracji sterującej ogrzewaniem lub chłodzeniem w zależności od podanego czujnika temperatury i wprowadzonego harmonogramu ogrzewania bądź chłodzenia w rozkładzie tygodniowym. Wartości temperatury ustawiane są przy pomocy interfejsu graficznego dla każdego dnia i godziny z rozdzielczością 15-minutową, 30-minutową lub godzinną.
W jednym CLU można utworzyć do 64 termostatów.

> **Uwaga!** Po utworzeniu termostatu (po wysłaniu nowej konfiguracji do CLU) staje się on automatycznie aktywny. Chcąc zatrzymać jego pracę, należy wywołać metodę `Stop`.

A. Tworzenie termostatu

W celu utworzenia termostatu należy zaznaczyć CLU, w ramach którego ma zostać umieszczony, a następnie z menu górnego uruchomić `Dodaj obiekt CLU`.

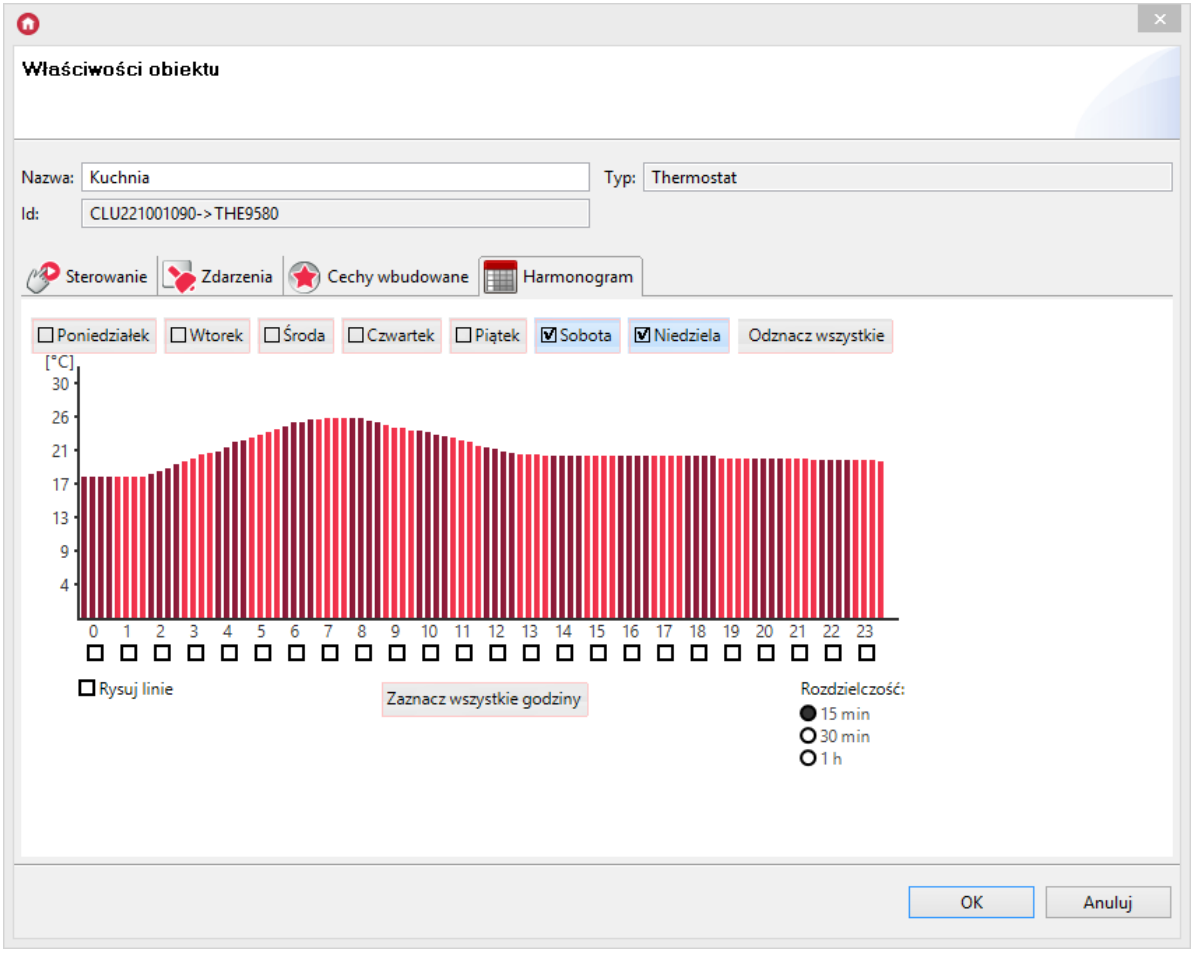


W otwartym oknie wyboru należy odszukać i wybrać `Thermostat`. Po wpisaniu nazwy, wprowadzeniu źródła (którym powinien być czujnik temperatury odpowiedzialny za daną strefę grzewczą) oraz wyborze odbiornika (będącego wyjściem, do którego podłączone jest urządzenie odpowiedzialne za daną strefę grzewczą - np. głowica kaloryfera, ogrzewanie podłogowe) dla utworzonego obiektu na ekranie otworzy się okno właściwości harmonogramu. W oknie tym znajdują się cztery zakładki:

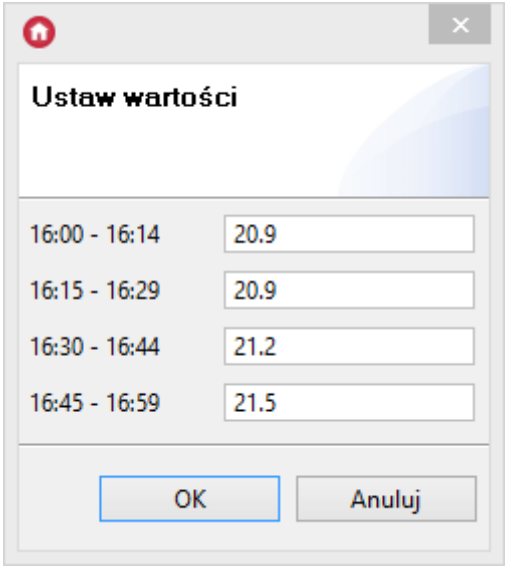
- **Sterowanie** - zawiera metody harmonogramu;
- **Zdarzenia** - zawiera zdarzenia harmonogramu;
- **Cechy wbudowane** - zawiera listę cech harmonogramu;
- **Harmonogram** - zawiera interfejs graficzny, umożliwiający proste formułowanie wartości dla całego zakresu harmonogramu.

B. Formułowanie wartości dla termostatu

W zakładce *Harmonogram* (w oknie właściwości) znajduje się interfejs graficzny, dzięki któremu istnieje możliwość zadawania wartości.



Harmonogram pozwala na wpisanie wartości dla 7 dni (w zakresie jednego tygodnia) z rozdzielczością 15-minutową. Można ustawić wartości dla każdego dnia z osobna lub dla kilku dni jednocześnie. Dzień, dla którego aktualnie wpisywane są wartości, wyróżniony jest czarnym znacznikiem z lewej strony nazwy. Przełączenie na inny dzień następuje po kliknięciu na jego nazwę.
W celu jednoczesnego wprowadzania wartości dla kilku dni, należy kliknąć znaczniki przy nazwach, dla których wartości będą ustawiane. Wartości można ustawiać bezpośrednio na wykresie posługując się myszką lub wpisywać ręcznie w oknie wartości, które otwiera się po kliknięciu na wybraną godzinę.



Termostat reaguje na harmonogram, gdy znajduje się w trybie `Auto`. Wybór trybu pracy odbywa się przy pomocy aplikacji lub przez metody obiektu.

C. Parametry konfiguracyjne obiektu Termostat

CECHY

Nazwa	Opis
`Source`	Wejście termostatu, powiązanie z czujnikiem temperatury
`Control`	Wyjście termostatu, powiązanie z elementem wykonującym
`OutputType`	Określenie typu wyjścia (-1 - autodetekcja, 0 - wyjście cyfrowe, 1 - wyjście analogowe)
`PointValue`	Wartość temperatury zadana ręcznie
`HolidayModeValue`	Wartość temperatury dla trybu wakacyjnego
`Hysteresis`	Wartość histerezy - określenie granic załączenia i wyłączenia termostatu
`State`	Stan pracy (1 - termostat aktywny, 0 - nieaktywny)
`ControlDirection`	Kierunek pracy (0 - tryb normalny (grzanie), 1 - tryb odwrotny (chłodzenie))
`Mode`	Tryb pracy (0 - tryb ręczny (wykorzystanie `PointValue`), 1 - tryb wakacyjny (`HolidayModeValue`), 2 - tryb automatyczny (wartość `AutoMode` z Harmonogramu), 3 - tryb podgrzewania (wartość `HeatUp`))
`Data`	Ciąg znaków definiujący harmonogram zmian wartości
`Min`	Dolna wartość zakresu wbudowanego harmonogramu
`Max`	Górna wartość zakresu wbudowanego harmonogramu
`TargetTemp`	Aktualna wartość temperatury docelowej
`ControlOutValue`	Wartość przypisana do wyjścia sterującego ogrzewaniem

METODY

Nazwa	Opis
`Start`	Przełączenie termostatu w stan aktywny (`State`=1)
`Stop`	Przełączenie termostatu w stan nieaktywny (`State`=0)
`IncreaseDegree`	Zwiększenie `PointValue` o 1°C
`DecreaseDegree`	Zmniejszenie `PointValue` o 1°C
`HeatUp`	Zwiększenie `PointValue` o zadaną wartość w określonym czasie
`HolidayModeStart`	Uruchomienie trybu wakacyjnego
`HolidayModeStop`	Zatrzymanie trybu wakacyjnego
`AutoModeStart`	Uruchomienie trybu `AutoMode` (pobieranie temperatury z harmonogramu)
`AutoModeStop`	Zatrzymanie trybu `AutoMode`
`SetData`	Ustawienie tygodniowego harmonogramu
`SetOutputType`	Ustawienie typu wyjścia (`Auto` - autodetekcja, `Digital` - wyjście cyfrowe, `Analog` - wyjście analogowe)
`SetPointValue`	Ustawienie ręcznie zadanej temperatury
`SetHolidayModeValue`	Ustawienie wartości temperatury dla trybu wakacyjnego
`SetHysteresis`	Ustawienie wartości histerezy
`SetControlDirection`	Ustawienie kierunku pracy (0 - tryb normalny (grzanie), 1 - tryb odwrotny (chłodzenie))

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie generowane przy zmianie wartości cechy `PointValue`
`OnStart`	Zdarzenie generowane przy wznowieniu pracy termostatu
`OnStop`	Zdarzenie generowane przy zatrzymaniu pracy termostatu
`OnOutOn`	Zdarzenie wywoływane przy wystawieniu na wyjściu `OutValue` wartości większej od zera
`OnOutOff`	Zdarzenie wywoływane przy wystawieniu na wyjściu `OutValue` wartości mniejszej od zera
`OnHolidayModeOn`	Zdarzenie generowane przy uruchomieniu trybu wakacyjnego
`OnHolidayModeOff`	Zdarzenie generowane przy wyłączeniu trybu wakacyjnego

6. Push

Powiadomienia typu Push są wirtualnymi obiektami tworzonymi przez użytkownika w CLU. Umożliwiają one wysyłanie powiadomień na urządzenie, które posiada zainstalowaną aplikację myGrenton. Możliwe jest utworzenie maksymalnie 64 obiektów Push na jednym CLU.

> **Uwaga!** Aby korzystać z funkcjonalności powiadomień Push, należy posiadać Object Manager v1.3.3 (lub wyższy) oraz aplikację (Android) myGrenton v1.1.9 (lub wyższa) lub (iOS) myGrenton v1.3.0 (lub wyższa). Dodatkowo w CLUZ musi być załączone korzystanie z chmury (Cechy wbudowane `UseCloud` oraz `CloudConnection` muszą mieć wartość `true`).

A. Tworzenie powiadomienia Push

W celu utworzenia powiadomienia Push należy zaznaczyć CLU, w ramach którego ma się znaleźć, a następnie z menu górnego uruchomić `Dodaj obiekt CLU`.



W otwartym oknie wyboru należy odszukać i wybrać obiekt `Push`, a następnie nadać mu nazwę.
Na ekranie pojawi się okno właściwości nowo utworzonego powiadomienia Push, w którym znajdują się trzy zakładki:

- **Sterowanie** – zawiera metody powiadomienia Push;
- **Zdarzenia** – zawiera zdarzenia powiadomienia Push;
- **Cechy wbudowane** – zawiera listę cech powiadomienia Push.

B. Parametry konfiguracyjne obiektu Push

CECHY

Nazwa	Opis
`Message`	Treść notyfikacji Push
`Title`	Tytuł notyfikacji Push
`LastSendTime`	Czas, w którym została wysłana ostatnia notyfikacja Push
`Interval`	Czas pomiędzy kolejnymi notyfikacjami (w sekundach)

METODY

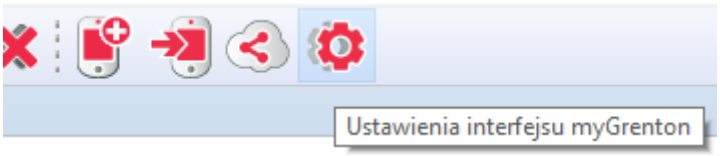
Nazwa	Opis
`SetMessage`	Dodaje treść notyfikacji Push
`ClearMessage`	Czyści treść notyfikacji Push
`SetTitle`	Dodaje tytuł notyfikacji Push
`ClearTitle`	Czyści tytuł notyfikacji Push
`Send`	Wysyła powiadomienie na urządzenie
`SetInterval`	Ustawia czas pomiędzy kolejnymi notyfikacjami (w sekundach)

ZDARZENIA

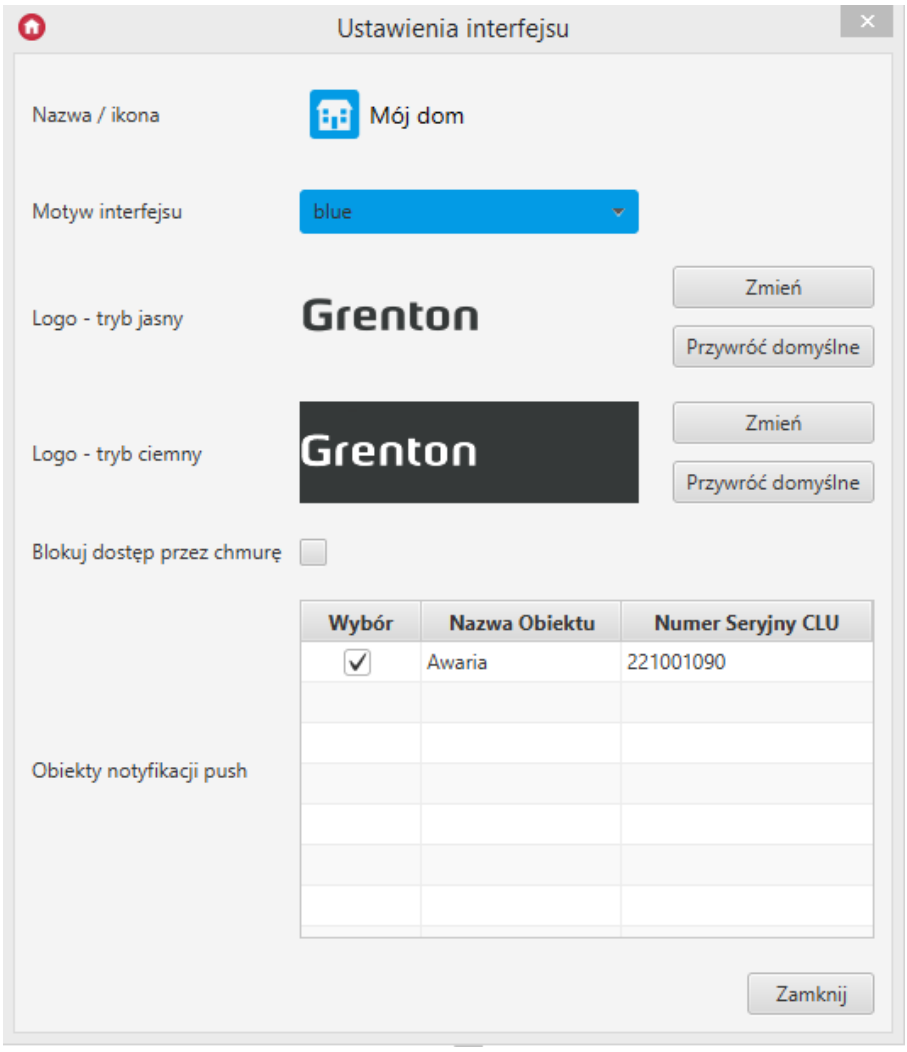
Nazwa	Opis
`OnSend`	Zdarzenie wywoływane przy wysłaniu notyfikacji
`OnOverflow`	Zdarzenie wywoływane przy przepełnieniu kolejki

C. Konfiguracja po stronie aplikacji myGrenton

Aby dodać powiadomienie do interfejsu myGrenton należy kliknąć ikonę `Ustawienia interfejsu myGrenton` w pasku narzędzi:



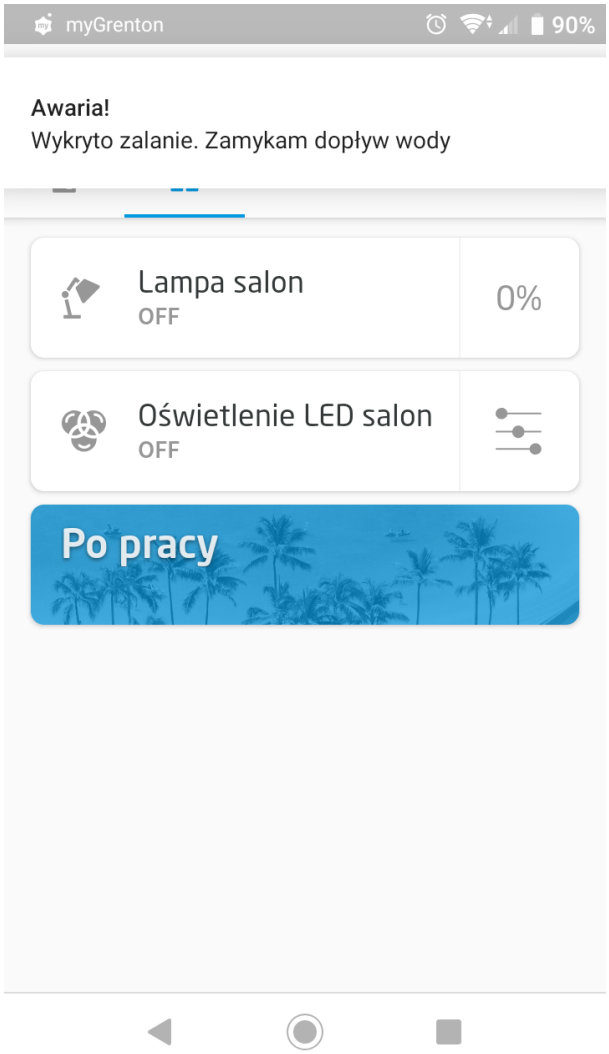
Po kliknięciu ikony pojawi się okno z ustawieniami interfejsu:



W pozycji `Obiekty notyfikacji push` należy wybrać notyfikacje, które chcemy aktywować w danym interfejsie myGrenton, a następnie wysłać interfejs na urządzenie patrz pkt XVII.5..

D. Działanie powiadomień Push

Po poprawnym ustawieniu i wysłaniu interfejsu na urządzenie, wywołanie metody `Send` powoduje pojawienie się powiadomienia Push na ekranie urządzenia.



Wysłanie większej ilości powiadomień z jednego obiektu Push skutkuje dodaniem ich do kolejki i pojawianiem się na urządzeniu w odstępach czasu określonych cechą `Interval`.

WŁAŚCIWOŚCI KOLEJKI

- Jednocześnie w kolejce do wysłania może znajdować się maksymalnie 10 wiadomości;
- Jeśli w kolejce pojawi się więcej niż 10 wiadomości, wygenerowane zostanie zdarzenie przepełnienia kolejki `OnOverflow`;
- Jeśli w kolejce pojawi się więcej niż 10 wiadomości, na urządzenie zostanie wysłane ostatnie 10 wiadomości.

> **Uwaga!** Wiadomości wysłane z różnych obiektów Push dotrą do urządzenia jednocześnie.

7. Czujnik obecności

Czujnik obecności (PresenceSensor) jest wirtualnym obiektem tworzonym w ramach danego CLU. Object Manager umożliwia utworzenie maksymalnie 64 obiektów. Czujnik obecności może być wykorzystywany wszędzie tam, gdzie potrzebne jest wywołanie metody po określonym czasie, jak również zliczanie czasu od ostatniego wywołania (np. po detekcji ruchu).

> **Uwaga!** Aby w pełni korzystać z opisanej funkcjonalności obiektu Presence Sensor, należy posiadać CLUZ z firmware 5.10.01 lub wyższym.

A. Tworzenie czujnika obecności

W celu utworzenia czujnika obecności należy zaznaczyć CLU, w ramach którego ma się znaleźć, a następnie z menu górnego uruchomić `Dodaj obiekt CLU`.



Po kliknięciu ikony pojawia się okno wyboru z listą dostępnych obiektów, na której należy odszukać i wybrać obiekt `PresenceSenor`. Po zaznaczeniu, wciśnięciu `OK`, konieczne jest nadanie nazwy nowemu czujnika obecności. Utworzony czujnik pojawi się na liście obiektów zaznaczonego CLU.

B. Tryb pracy czujnika obecności

Czujnik obecności może pracować w dwóch trybach (`mode`) w zależności od użytego czujnika ruchu:

- **Impulse input** - tryb, w którym po wykryciu ruchu przez czujnik cecha `PresenceDetected` = 1 podtrzymywana jest w czasie `Timeout` - po upływie czasu cecha `PresenceDetected` zmienia stan na 0;
- **State input** - tryb, w którym po wykryciu ruchu przez czujnik cecha `PresenceDetected` = 1 podtrzymywana jest do czasu uruchomienia metody `UndetectPresence` - po jej uruchomieniu stan wysoki cechy `PresenceDetected` utrzymywany jest w czasie `Timeout`.

C. Parametry konfiguracyjne czujnika obecności

CECHY

Nazwa	Opis
`Timeout`	Czas (w sekundach) od ostatniej aktywności, po którym wartość cechy PresenceDetected zostaje ustawiona na 0
`State`	Aktualny stan czujnika obecności: 1 - włączony 0 - wyłączony
`PresenceDetected`	Wartość mówiąca o wykryciu ruchu przez czujnik
`TimeFromLastPresence`	Czas od ostatniego wykrycia ruchu (z czujnika lub włączenia światła z przycisku). Resetowany po wywołaniu: - DetectPresence() bez względu na Locked, DetectionDelay - SwitchLocked() przy zmianie Locked na 1 - SetLocked(On) W trybie stanowym po wywołaniu DetectPresence(), TimeFromLastPresence = 0, aż do wywołania UndetectPresence()
`DetectionDelay`	Czas ignorowania DetectPresence po zmianie Locked On->Off
`Locked`	Stan zablokowania obecności: 0 - reagowanie na DetectPresence 1 - podtrzymanie PresenceDetected jako 1
`Mode`	Tryb działania obiektu w zależności od typu używanego czujnika ruchu: 0 - impulsowy 1 - stanowy

METODY

Nazwa	Opis
`Start`	Uruchamia czujnik obecności
`Stop`	Zatrzymuje czujnik obecności
`DetectPresence`	Metoda wywoływana podczas wykrycia obecności. Ustawia wartość cechy PresenceDetected na 1 oraz resetuje licznik TimeFromLastPresence czujnika obecności
`UndetectPresence`	Używana w trybie stanowym (Mode = 1). Kończy podtrzymywanie parametru PresenceDetected (po upływie Timeout'u)
`SwitchLocked`	Zmienia wartość parametru Locked na przeciwny. Przypadki: - zmiana Locked z 0 na 1 - ustawienie na 1 i zablokowanie PresenceDetected, wywołanie zdarzenia OnSwitchOn (jeśli wcześniej PresenceDetected = 0), reset TimeFromLastPresence - zmiana Locked z 1 na 0 - ustawienie na 0 i odblokowanie PresenceDetected, wywołanie zdarzenia OnSwitchOff
`SetLocked`	Ustawia wartość parametru Locked. Przypadki: - zmiana Locked z 0 na 1 - ustawienie na 1 i zablokowanie PresenceDetected, wywołanie zdarzenia OnSwitchOn (jeśli wcześniej PresenceDetected = 0), reset TimeFromLastPresence - zmiana Locked z 1 na 0 - ustawienie na 0 i odblokowanie PresenceDetected, wywołanie zdarzenia OnSwitchOff - SetLocked(On) jeśli Locked = 1 - resetuje TimeFromLastPresence - SetLocked(Off) jeśli Locked = 0 - brak reakcji
`SetTimeout`	Ustawia parametr Timeout (wyrażony w sekundach)
`SetDetectionDelay`	Ustawia parametr DetectionDelay (wyrażony w sekundach)
`SetMode`	Ustawia parametr Mode

ZDARZENIA

Nazwa	Opis
`OnStart`	Zdarzenie wywoływane przy uruchomieniu czujnika
`OnStop`	Zdarzenie wywoływane przy dezaktywacji czujnika
`OnSwitchOn`	Zdarzenie wywoływane podczas wykrycia obecności (zmiana wartości parametru PresenceDetected z 0 na 1)
`OnSwitchOff`	Zdarzenie wywoływane na timeout licznika (zmiana parametru PresenceDetected z 1 na 0)

8. Kalendarz wschodów i zachodów słońca

Kalendarz wschodów i zachodów słońca (Sunrise Sunset Calendar) jest wirtualnym obiektem tworzonym w ramach danego CLU. Object Manager umożliwia utworzenie maksymalnie 64 obiektów. Kalendarz może być wykorzystywany w sytuacji, gdy dane akcje w systemie mają być wykonywane w zależności od pory dnia (wschód / zachód słońca).

> **Uwaga!** Aby w pełni korzystać z opisanej funkcjonalności obiektu kalendarza wschodu / zachodu słońca, należy posiadać CLUZ z firmware 5.09.02 lub wyższym.

A. Tworzenie kalendarza

W celu utworzenia kalendarza wschodów i zachodów słońca należy zaznaczyć CLU, w ramach którego ma się znaleźć, a następnie z menu górnego uruchomić `Dodaj obiekt CLU`.



Po kliknięciu ikony pojawia się okno wyboru z listą dostępnych obiektów, na której należy odszukać i wybrać obiekt `SunriseSunsetCalendar`. Po zaznaczeniu, wciśnięciu `OK`, konieczne jest nadanie nazwy nowego kalendarza. Utworzony kalendarz pojawi się na liście obiektów zaznaczonego CLU.

B. Parametry konfiguracyjne kalendarza

CECHY

Nazwa	Opis
`Longitude`	Długość geograficzna w stopniach dziesiętnych (DD), zakres -180 do 180
`Latitude`	Szerokość geograficzna w stopniach dziesiętnych (DD), zakres -90 do 90
`State`	Aktualny stan kalendarza wschodów i zachodów słońca: 1 - włączony, 0 - wyłączony
`SunriseUTC`	Czas wschodu słońca dla ustawionej lokalizacji w UTC (± 5 minut) N\A - Brak możliwości wyliczenia wschodu słońca
`SunsetUTC`	Czas zachodu słońca dla ustawionej lokalizacji w UTC (± 5 minut) N\A - Brak możliwości wyliczenia zachodu słońca
`SunriseLocal`	Lokalny czas wschodu słońca dla ustawionej lokalizacji (± 5 minut) N\A - Brak możliwości wyliczenia wschodu słońca dla ustawionej lokalizacji
`SunsetLocal`	Lokalny czas zachodu słońca dla ustawionej lokalizacji (± 5 minut) N\A - Brak możliwości wyliczenia zachodu słońca dla ustawionej lokalizacji
`SunriseUTCTimestamp`	Czas wschodu słońca dla ustawionej lokalizacji w UTC (± 300 sekund) -1 - Brak możliwości wyliczenia wschodu słońca
`SunsetUTCTimestamp`	Czas zachodu słońca dla ustawionej lokalizacji w UTC (± 300 sekund) -1 - Brak możliwości wyliczenia zachodu słońca
`SunriseLocalTimestamp`	Lokalny czas wschodu słońca dla ustawionej lokalizacji (± 300 sekund) -1 - Brak możliwości wyliczenia wschodu słońca dla ustawionej lokalizacji
`SunsetLocalTimestamp`	Lokalny czas zachodu słońca dla ustawionej lokalizacji (± 300 sekund) -1 - Brak możliwości wyliczenia zachodu słońca dla ustawionej lokalizacji
`IsDayNow`	Określa aktualną część doby na podstawie lokalnego wschodu/zachodu słońca (± 5 minut) - nie uwzględnia offsetów SunriseOffset / SunsetOffset: 0 - noc, 1 - dzień, -1 - Brak możliwości wyliczenia dla ustawionej lokalizacji
`SunriseOffset`	Przesunięcie zdarzenia wschodu słońca OnSunrise (w minutach), zakres: -1439 do 1439
`SunsetOffset`	Ustawia przesunięcie zdarzenia OnSunset, wartość cechy SunsetOffset(w minutach), zakres: -1439 do 1439

Nazwa	Opis
`NextSunrise`	Pozostały czas do wschodu słońca (w minutach) -1 - Brak możliwości wyliczenia dla ustawionej lokalizacji
`NextSunset`	Pozostały czas do zachodu słońca (w minutach) -1 - Brak możliwości wyliczenia dla ustawionej lokalizacji

METODY

Nazwa	Opis
`Start`	Uruchamia kalendarz wschodów i zachodów słońca
`Stop`	Zatrzymuje kalendarz wschodów i zachodów słońca
`SetLongitude`	Ustawia długość geograficzną w stopniach dziesiętnych (DD), zakres -180 do 180
`SetLatitude`	Ustawia szerokość geograficzną w stopniach dziesiętnych (DD), zakres -90 do 90
`SetSunriseOffset`	Ustawia przesunięcie zdarzenia OnSunrise, wartość cechy SunriseOffset, zakres: -1439 do 1439
`SetSunsetOffset`	Ustawia przesunięcie zdarzenia OnSunset, wartość cechy SunsetOffset, zakres: -1439 do 1439

ZDARZENIA

Nazwa	Opis
`OnStart`	Zdarzenie wywoływane przy uruchomieniu kalendarza
`OnStop`	Zdarzenie wywoływane przy dezaktywacji kalendarza
`OnSunrise`	Zdarzenie wywoływane podczas wschodu słońca z uwzględnieniem offsetu
`OnSunset`	Zdarzenie wywoływane podczas zachodu słońca z uwzględnieniem offsetu
`OnSunriseSunsetChange`	Zdarzenie wywoływane podczas wschodu lub zachodu słońca z uwzględnieniem offsetu
`OnDay`	Zdarzenie wywoływane podczas zmiany cechy IsDayNow z 0 (noc) na 1 (dzień)
`OnNight`	Zdarzenie wywoływane podczas zmiany cechy IsDayNow z 1 (dzień) na 0 (noc)

9. Harmonogram zdarzeń

Harmonogram zdarzeń (Event Scheduler) to obiekt wirtualny tworzony w ramach danego CLU. Służy do wywoływania zadanego zdarzenia dla danej godziny oraz dnia tygodnia. W celu określenia godziny oraz dnia tygodnia, w której zdarzenie ma zostać wykonane, należy dodać odpowiednią regułę za pomocą metody `AddRule`. Reguły tworzone są za pomocą składni zgodnej z regułami CRON systemu LINUX.

> **Uwaga!** Aby w pełni korzystać z opisanej funkcjonalności harmonogramu zdarzeń, należy posiadać CLUZ z firmware 5.09.02 lub wyższym.

Do konfiguracji oraz zarządzania reguł zalecane jest wykorzystywanie widgetu `EVENT_SCHEDULER` z poziomu aplikacji myGrenton - patrz pkt XVIII.3.18.

A. Tworzenie harmonogramu zdarzeń

W celu utworzenia harmonogramu zdarzeń należy zaznaczyć CLU, w ramach którego ma się znaleźć, a następnie z menu górnego uruchomić `Dodaj obiekt CLU`.



Po kliknięciu ikony pojawia się okno wyboru z listą dostępnych obiektów, na której należy odszukać i wybrać obiekt `EventScheduler`. Po zaznaczeniu, wciśnięciu `OK`, konieczne jest nadanie nazwy nowego harmonogramu zdarzeń. Utworzony harmonogram zdarzeń pojawi się na liście obiektów zaznaczonego CLU.

B. Reguły harmonogramu zdarzeń

Dodawanie reguł do harmonogramu odbywa się na dwa sposoby:

- za pomocą aplikacji myGrenton i widżetu `EVENT_SCHEDULER` -> szczegółowy opis można znaleźć w rozdziale patrz pkt XVIII.3.18.
- poprzez wpisanie reguły CRON z wykorzystaniem metody `AddRule` w zakładce sterowanie - szczegółowe informacje na temat sposobu tworzenia reguł w formacie CRON można znaleźć w dokumentacji kalendarza CRON.

> **Uwaga!** Obiekt wirtualny Harmonogram zdarzeń korzysta jedynie z trzech członów zapisu CRON "**minuta godzina** dzień miesiąc **dzień_tygodnia**". Wartości dnia i miesiąca są ignorowane, dlatego w miejsce tych wartości należy wprowadzić `*`. Docelowa składnia CRON wygląda następująco: `minuta godzina * * dzień_tygodnia`

C. Parametry konfiguracyjne harmonogramu zdarzeń

CECHY

Nazwa	Opis
`RuleList`	Lista wszystkich reguł w formacie `{{id, rule_state, "crone"}, {id, rule_state, "crone"},...}` `rule_state`: 0 - reguła wyłączona, 1 - reguła włączona
`CurrentRule`	Reguła z listy, która odpowiada za aktualne zdarzenie `{id, rule_state, "crone"}` Zwraca uruchomioną regułę przez 1 minutę, następnie "N\A" `rule_state`: 0 - reguła wyłączona, 1 - reguła włączona
`NextRule`	Reguła z listy dla następnego zdarzenia `{id, rule_state, "crone"}` `rule_state`: 0 - reguła wyłączona, 1 - reguła włączona
`RuleCount`	Liczba dodanych reguł
`RuleAvailableCount`	Liczba reguł, które mogą być dodane do istniejącej listy (wolne miejsce)
`State`	Aktualny stan harmonogramu zdarzeń

METODY

Nazwa	Opis
`Start`	Uruchamia harmonogram zdarzeń
`Stop`	Zatrzymuje harmonogram zdarzeń
`AddRule`	Dodaje regułę do listy. Format podawanej reguły crone `"minute hour * * day_of_week"`. Zwraca numer id przydzielonej reguły. Jeżeli metoda zwraca 0 - błąd
`DeleteRule`	Usuwa regułę o podanym id z listy. Metoda zwraca: 0 - ok, 1 - błąd
`GetRule`	Zwraca regułę w formacie `{id, rule_state, "crone"}` dla podanego id `rule_state`: 0 - reguła wyłączona, 1 - reguła włączona
`EnableRule`	Zmienia stan reguły na aktywny. Metoda: zwraca: 0 - ok, 1 - błąd
`DisableRule`	Zmienia stan reguły na nieaktywny. Metoda zwraca: 0 - ok, 1 - błąd
`GetRules`	Lista wszystkich reguł w formacie `{{id, rule_state, "crone"},{id, rule_state, "crone"},...}` `rule_state`: 0 - reguła wyłączona, 1 - reguła włączona
`GetNextRule`	Reguła z listy dla następnego zdarzenia `{id, rule_state, "crone"}` `rule_state`: 0 - reguła wyłączona, 1 - reguła włączona

ZDARZENIA

Nazwa	Opis
`OnStart`	Zdarzenie wywoływane przy uruchomieniu harmonogramu zdarzeń
`OnStop`	Zdarzenie wywoływane przy dezaktywacji harmonogramu zdarzeń
`OnEvent`	Zdarzenie docelowe wywołane na podstawie ustalonych reguł i aktualnym czasie urządzenia
`OnRuleAdd`	Zdarzenie wywołane podczas dodania reguły do listy
`OnRuleDelete`	Zdarzenie wywołane podczas usunięcia reguły z listy

X. Pomiar mediów

1. Wirtualny pomiar mediów

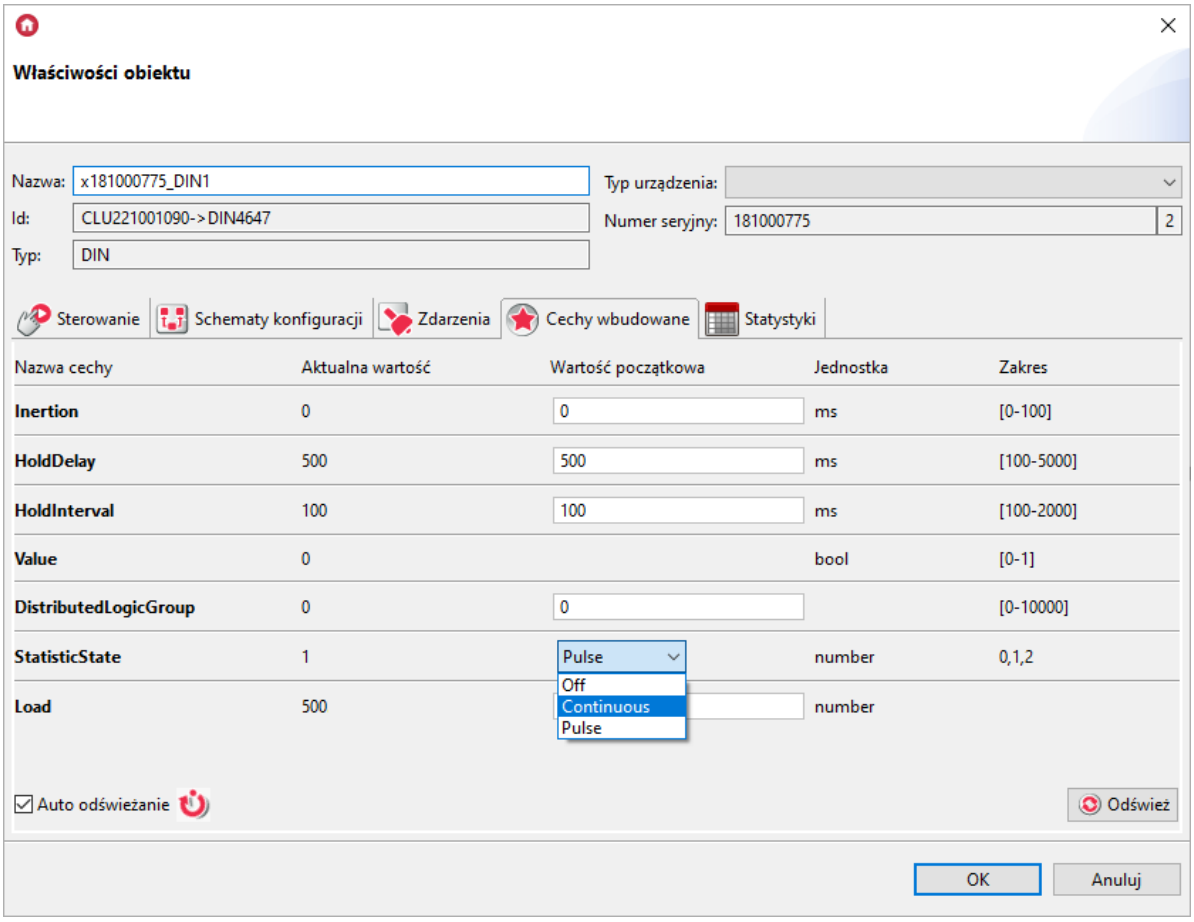
1.1. Uruchomienie wirtualnego pomiaru mediów po stronie Object Managera

Object Manager pozwala na przeprowadzenie pomiaru mediów, który umożliwia szacunkową prezentację zużytej energii (bazując na czasie załączenia urządzenia oraz podanej w konfiguracji mocy odbiornika). Konfiguracja pomiaru mediów odbywa się w OM i należy ją uruchomić dla każdego wejścia i wyjścia z osobna – tak, by CLU zbierało dane dotyczące zużycia energii. Pomiar mediów jest rejestrowany co 15 minut, rozpoczynając odliczanie od pełnej godziny - bazując na zegarze CLU (*cecha CLU-> TIME*).

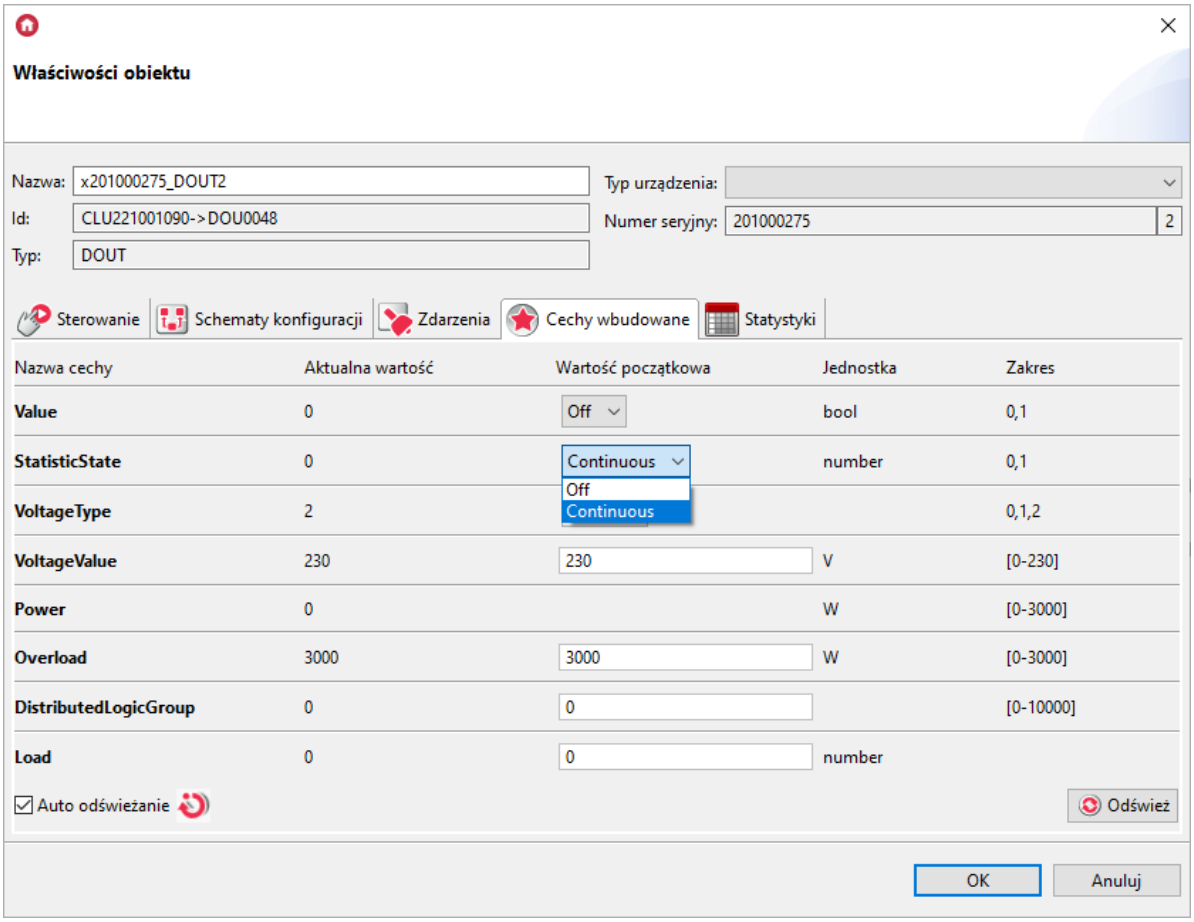
> **Uwaga!** Pomiar mediów dostępny jest dla Object Managera w wersji 1.2.0.180202 i wyższej oraz dla CLU z firmware 04.07.29-1802 i wyższym.

Pomiar mediów można uruchomić dla modułów:

- Wejściowych (Digital IN) – w trybie ciągłym (zliczającym czas pracy) lub impulsowym (zliczającym impulsy pojawiające się na wejściu binarnym):



- Wyjściowych (Relay, Led RGB, Dimmer) – w trybie ciągłym (zliczającym czas pracy):

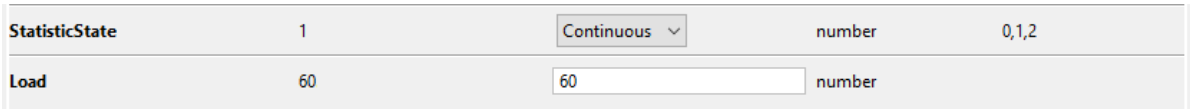


Uwaga! Pomiar mediów wyżej wymienionych modułów dotyczy modułów na szynę DIN oraz podtynkowych Tf-bus! Ustawienie pomiaru nie jest dostępne dla modułów Z-Wave!

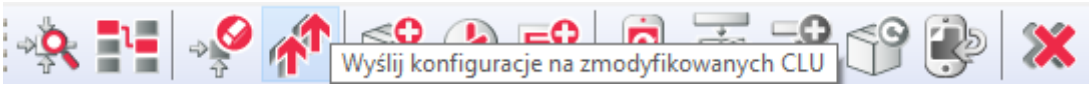
A. Stworzenie konfiguracji

Chcąc utworzyć konfigurację należy:

- Dwukrotnie kliknąć myszką na wybrany moduł z listy modułów w widoku głównym programu (dotyczy to wyżej wypisanych modułów umożliwiających obsługę pomiaru mediów);
- Przejść do zakładki *Cechy wbudowane*;
- Zmienić wybór cechy `StatisticState` na: `Continuous` (ciągły) lub `Pulse` (impulsowy) - w przypadku wejść binarnych modułu Digital In;
- Poniżej pojawi się cecha `Load` - do jej wartości początkowej wpisać pobieraną moc czynną urządzenia podłączonego do wejścia lub wyjścia w watach na godzinę (np. 60) - CLU podaną wartość przeliczy impulsowo lub ciągle (mnożąc przez czas w godzinach):



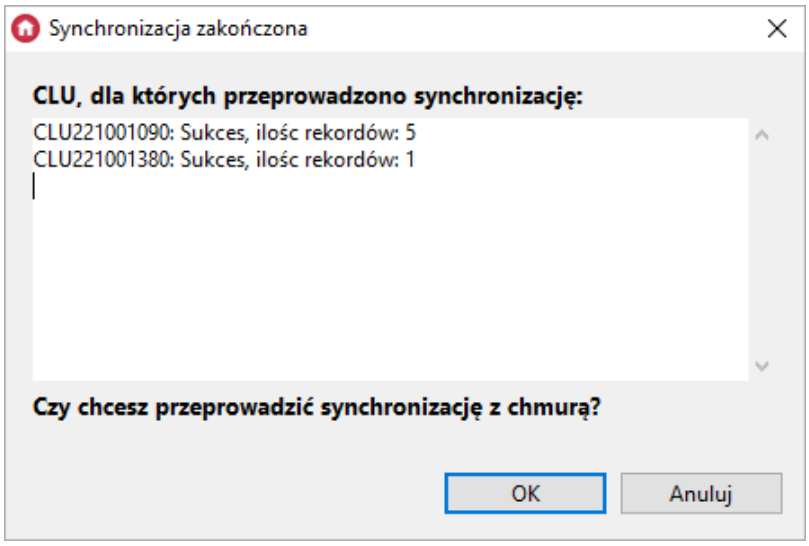
- Zatwierdzić przyciskiem *OK*;
- Dodać ustawienia pomiaru mediów dla kolejnych modułów - powtórzyć powyższe kroki;
- Wysłać konfigurację do CLU.



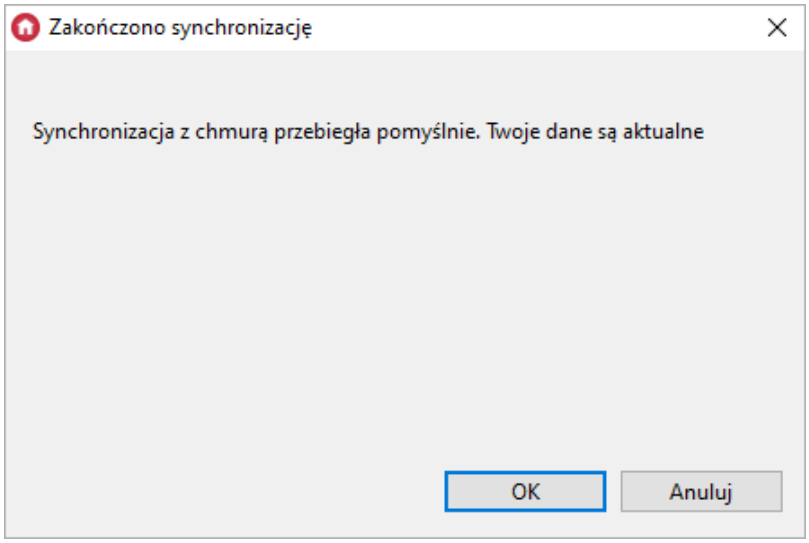
B. Odczyt pomiaru mediów w Object Managerze

W celu odczytania pomiaru mediów w programie Object Manager należy:

- Odczekać przynajmniej do pierwszego planowanego zarejestrowania pomiarów przez CLU (do XX.00 lub XX.15 lub XX.30 lub XX.45 - gdzie XX to godzina);
- Wybrać ***Narzędzia*** -> ***Pobierz plik z pomiarami***;
- Wyświetli się okno z informacją o pobranych rekordach:

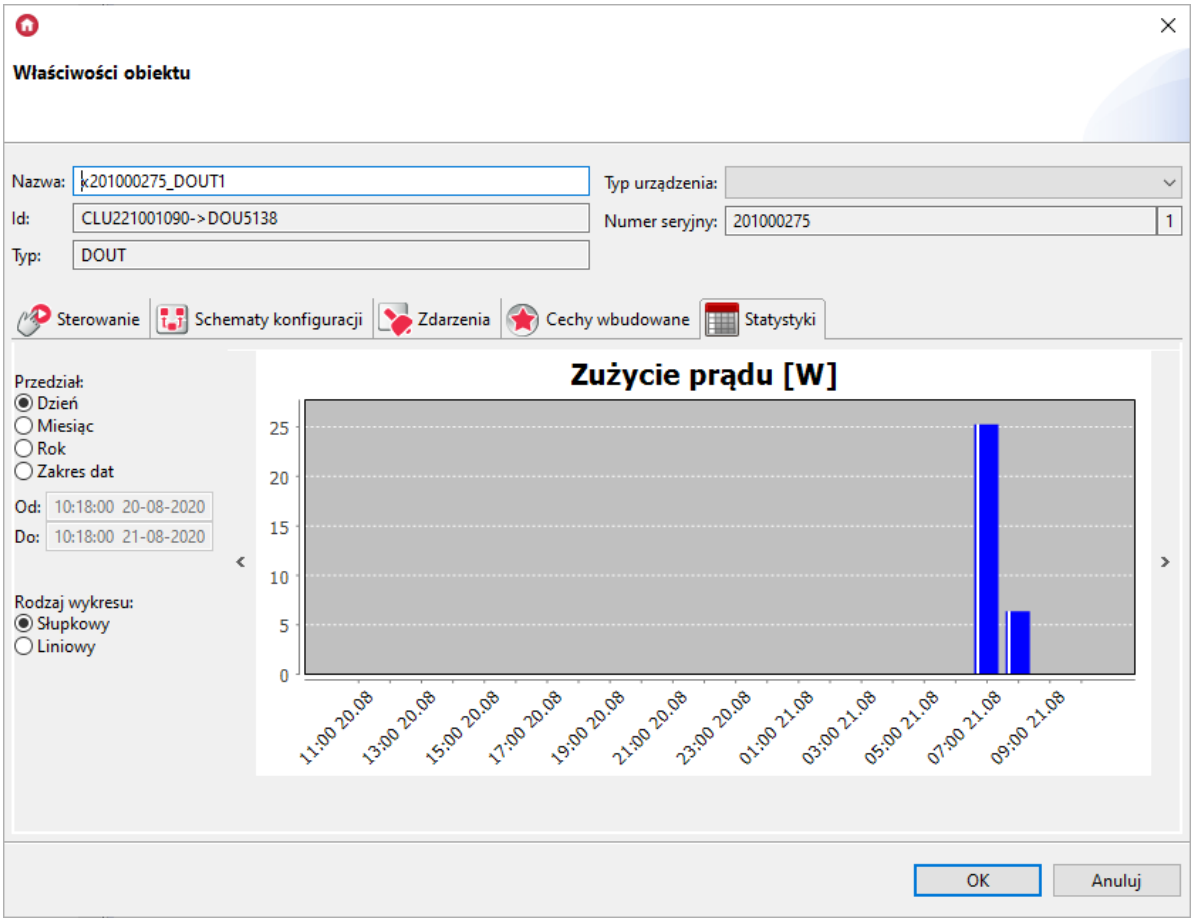


- Kliknąć *OK*;
- Object Manager następnie zsynchronizuje pobrane dane z chmurą;
- Po zakończonej synchronizacji wcisnąć *OK*:



> **Uwaga!** W przypadku błędu synchronizacji prosimy o kontakt z działem Support!

- W celu upewnienia się, czy pomiar mediów został zarejestrowany, należy kliknąć dwukrotnie na wybrany moduł, dla którego uruchomiony został pomiar mediów;
- Następnie należy przejść do zakładki *Statystyki*:
    - Można wybrać rodzaj wyświetlanego wykresu: słupkowy lub liniowy - w obu przypadkach na wykresie pojawia się zsumowana wartość zużytej energii (w watach) dla każdej godziny;
    - Można również wybrać przedział przeglądanego pomiaru mediów: dzień, miesiąc, rok lub samodzielnie wybrać zakres dat - w zależności od wybranego przedziału wyświetlony zostanie odpowiedni wykres.



C. Skonfigurowanie pomiaru mediów dla interfejsu aplikacji Home Manager

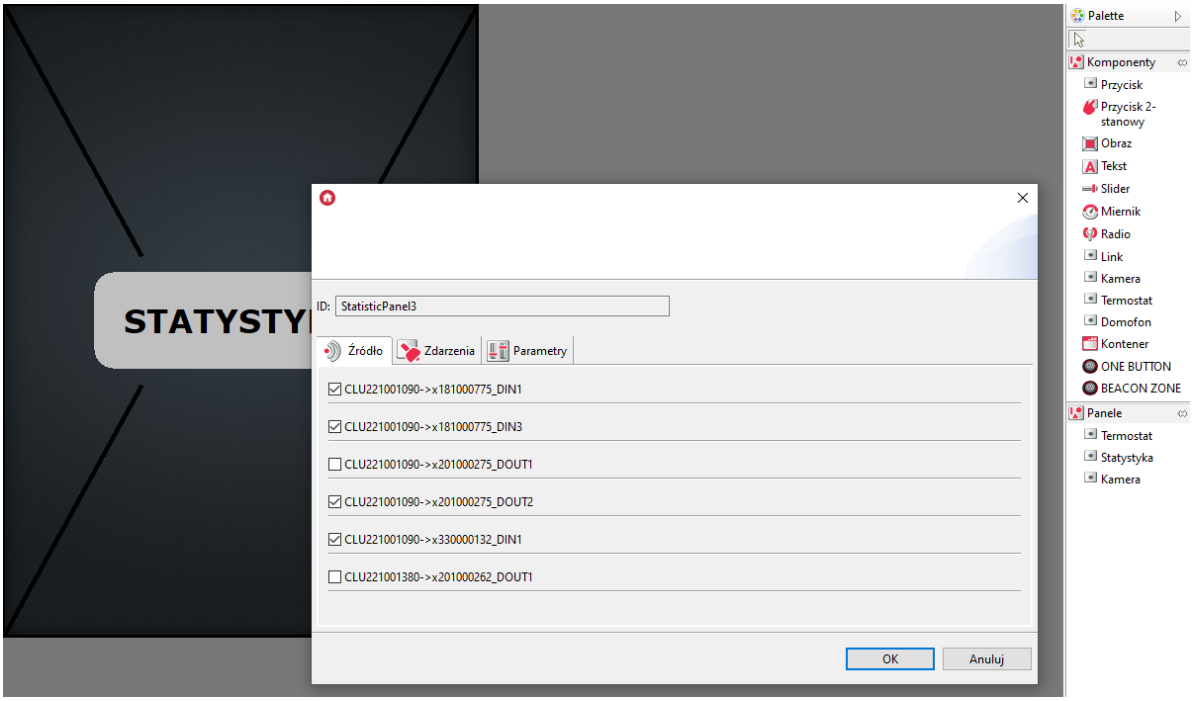
> **Uwaga!** Wsparcie dla funkcjonalności Visual Builder zostało wstrzymane. Kreator interfejsów Home Manager zostanie usunięty w następnych wersjach Object Manager.

Konfiguracja pomiaru mediów dla interfejsu aplikacji musi przebiegać zgodnie z poniższym schematem:

- Dodać nowy interfejs aplikacji:



- Wprowadzić nazwę tworzonej aplikacji;
- Ustawić rozdzielczość, skórkę, dodać przynajmniej jedną stronę, kliknąć *OK*;
- Z zasobnika paneli przeciągnąć panel *Statystyka* do obszaru edytowalnego interfejsu aplikacji;
- W zakładce *Źródło* zaznaczyć checkboxy dla modułów, których wykresy pomiaru mediów mają być wyświetlane w panelu statystyk w aplikacji:



- Kliknąć *OK*;
- Wysłać interfejs do urządzenia mobilnego - patrz pkt VIII.4.7..

1.2 Używanie pomiaru mediów po stronie aplikacji Home Manager

> **Uwaga!** Pomiar mediów dostępny jest dla Home Manager w wersji 1.1.110 lub wyższej.

By prawidłowo korzystać z pomiaru mediów w aplikacji mobilnej, należy w pierwszej kolejności pobrać pomiary z CLU oraz - jeśli jest to potrzebne - synchronizować pomiary z chmurą.

A. Pobieranie pomiarów

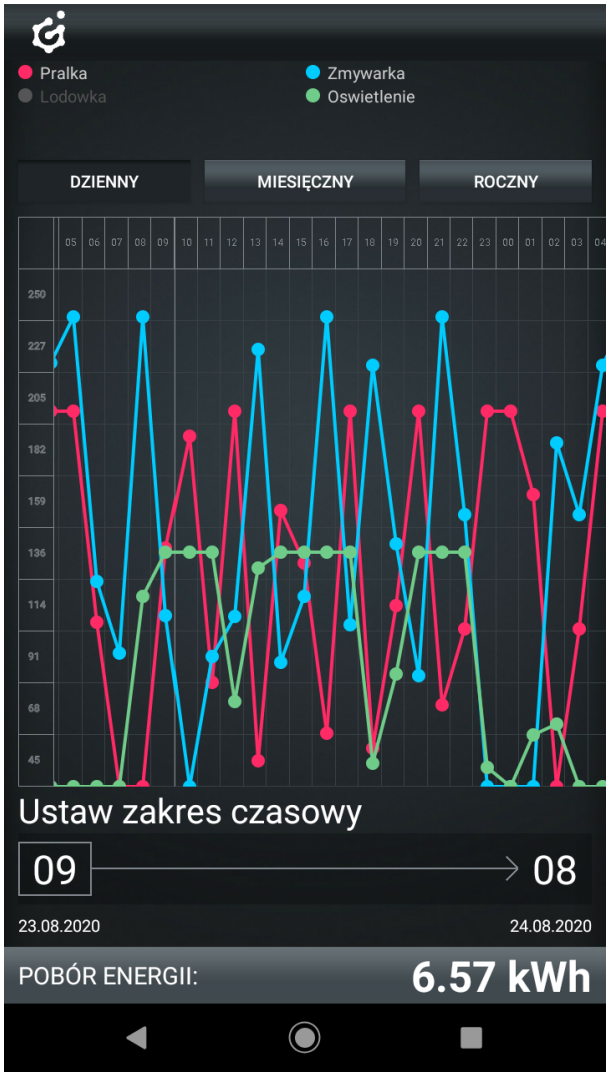
- Należy wejść do ustawień aplikacji z menu głównego (ikona koła zębatego).
- Z listy ustawień wybrać: *Pobierz pomiary z CLU*.
- Po chwili wyświetli się komunikat: *Sukces dla CLU: X, Y* [6].
- Uruchomić interfejs aplikacji – pomiary powinny być zaktualizowane i wyświetlone na wykresie.

B. Opcje widoku panelu pomiaru mediów

- Zmiana wyświetlanych danych konkretnych *wejść/wyjść* – po kliknięciu na wypisane moduły w górnej belce panelu pomiaru mediów, wyświetla się okno dostępnych modułów dodanych do panelu, które domyślnie są zaznaczone – ich odznaczenie powoduje brak pokazywania zmierzonych wartości dla konkretnych *wejść/wyjść*:



- W tym samym oknie, w którym są widoczne moduły, możliwa jest zmiana widoku wykresu - domyślne jest to wykres liniowy, można jednak również wybrać słupkowy, kołowy lub ranking;



- Zmiana zakresu czasowego wyświetlanych przebiegów – można tego dokonać przy pomocy przycisków *„dzienny”* (sumujący pomiary dla każdej godziny dnia), *„miesięczny”* (sumujący wartości dla każdego dnia w miesiącu) oraz *„roczny”* (sumujący pomiary dla każdego miesiąca z osobna);
- Możliwy jest również wybór własnego zakresu czasowego – po kliknięciu w daną godzinę wyświetla się okno wyboru dnia początkowego oraz końcowego:



C. Synchronizacja oraz pobieranie pomiarów

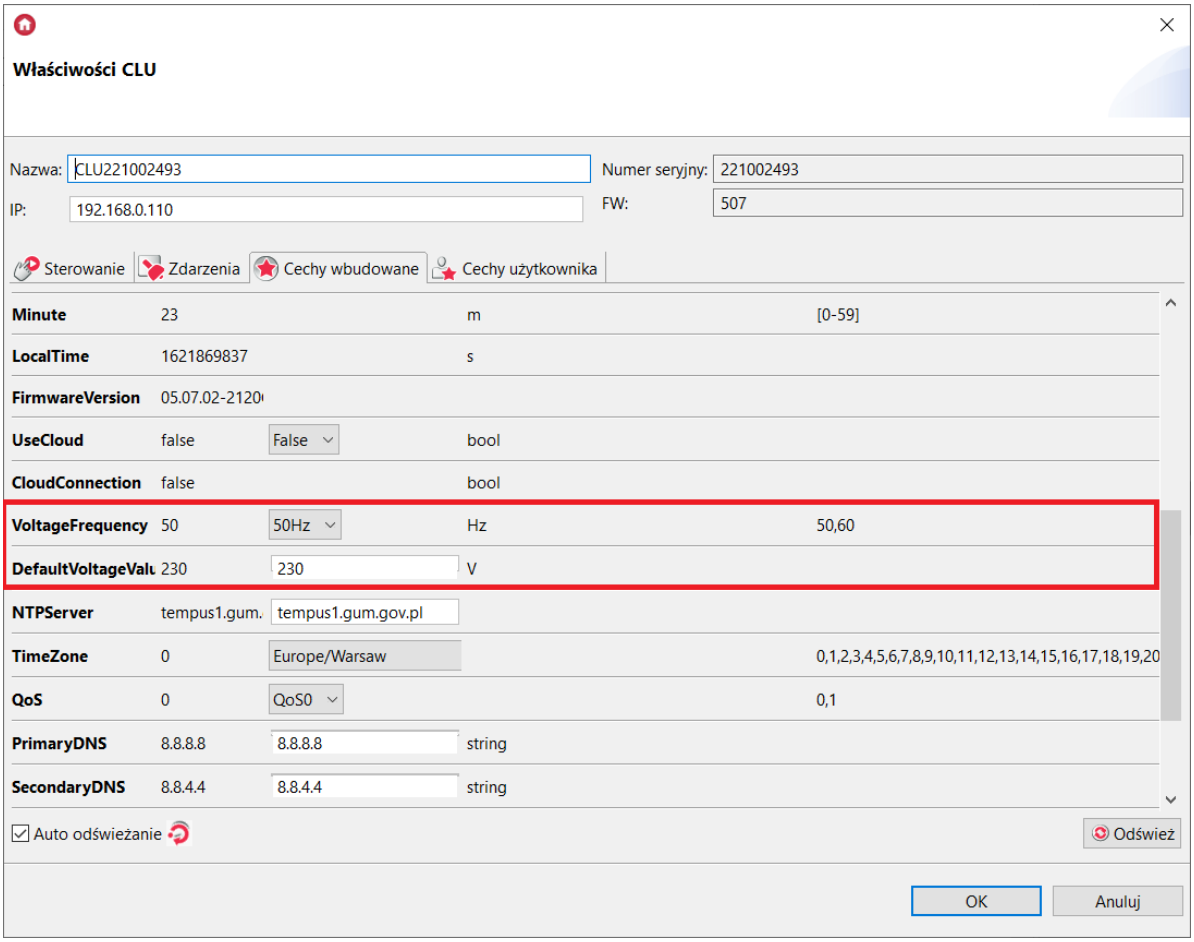
- Pobieranie pomiarów z CLU, które było dokonywane wcześniej, odbywało się przy połączeniu lokalnym z CLU. By pomiary mogły być wyświetlane podczas dostępu zdalnego, należy synchronizować je z chmurą;
- W celu zsynchronizowania pomiarów z chmurą należy wejść do menu głównego aplikacji Home Manager - w ustawienia i na samym dole wybrać: *Synchronizuj pomiary z chmurą*.

2. Rzeczywisty pomiar mediów

> **Uwaga!** Rzeczywisty pomiar mediów jest dostępny jedynie dla modułów z serii Grenton 2.0: GRENTON RELAY 2HP (DIN), GRENTON RELAY 4HP (DIN), GRENTON ROLLER SHUTTER (DIN), GRENTON ROLLER SHUTTER (Flush), GRENTON I/O MODULE 2/2 (Flush).

2.1. Ustawienia rzeczywistego pomiaru mediów po stronie Object Managera

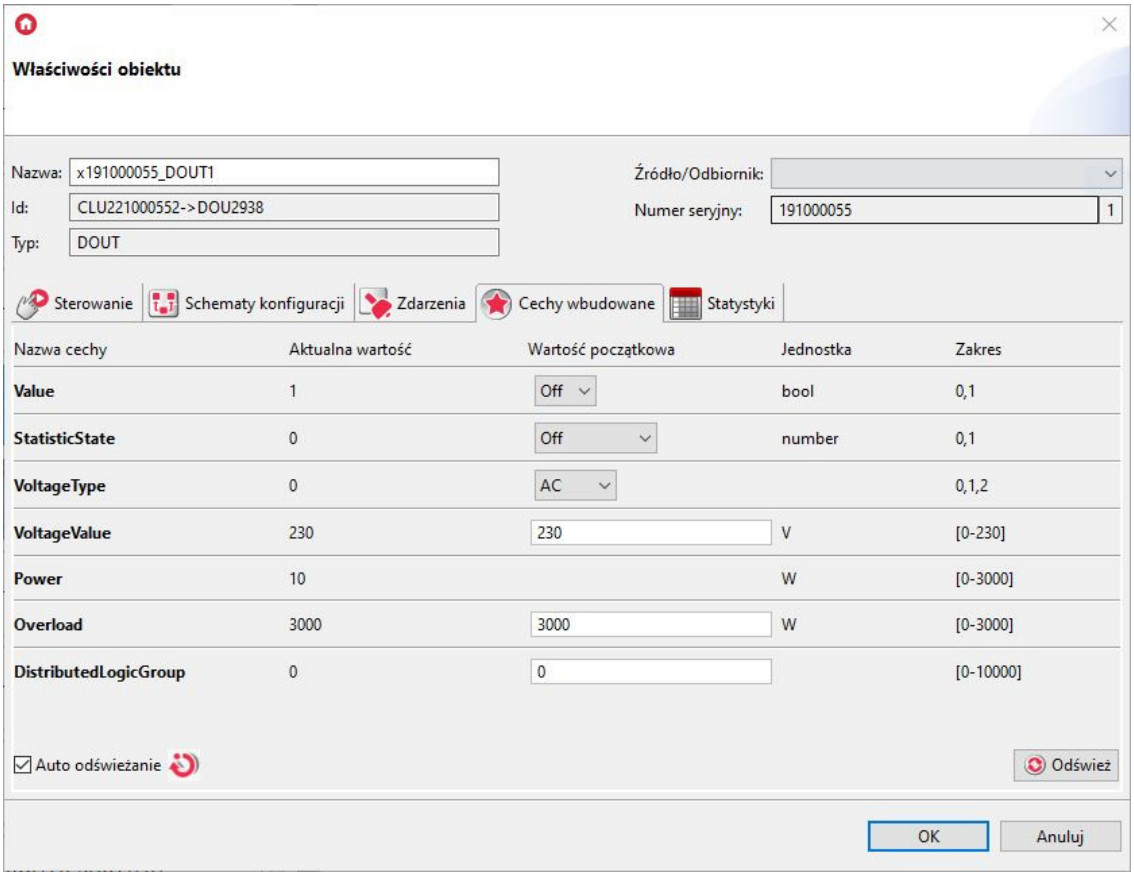
Object Manager pozwala na przeprowadzenie pomiaru mediów, który umożliwia rzeczywistą prezentację zużytej energii (bazując na parametrach `VoltageValue` oraz `VoltageType` urządzenia). Konfiguracja pomiaru mediów odbywa się w OM i należy ją sparametryzować dla każdego wyjścia z osobna. Aby pomiar mediów został przeprowadzony prawidłowo, należy określić parametry elektryczne sieci, do której podłączony jest układ. W tym celu, w cechach wbudowanych modułu CLU, należy zdefiniować częstotliwość (`VoltageFrequency`) oraz napięcie znamionowe (`DefaultVoltageValue`) sieci.



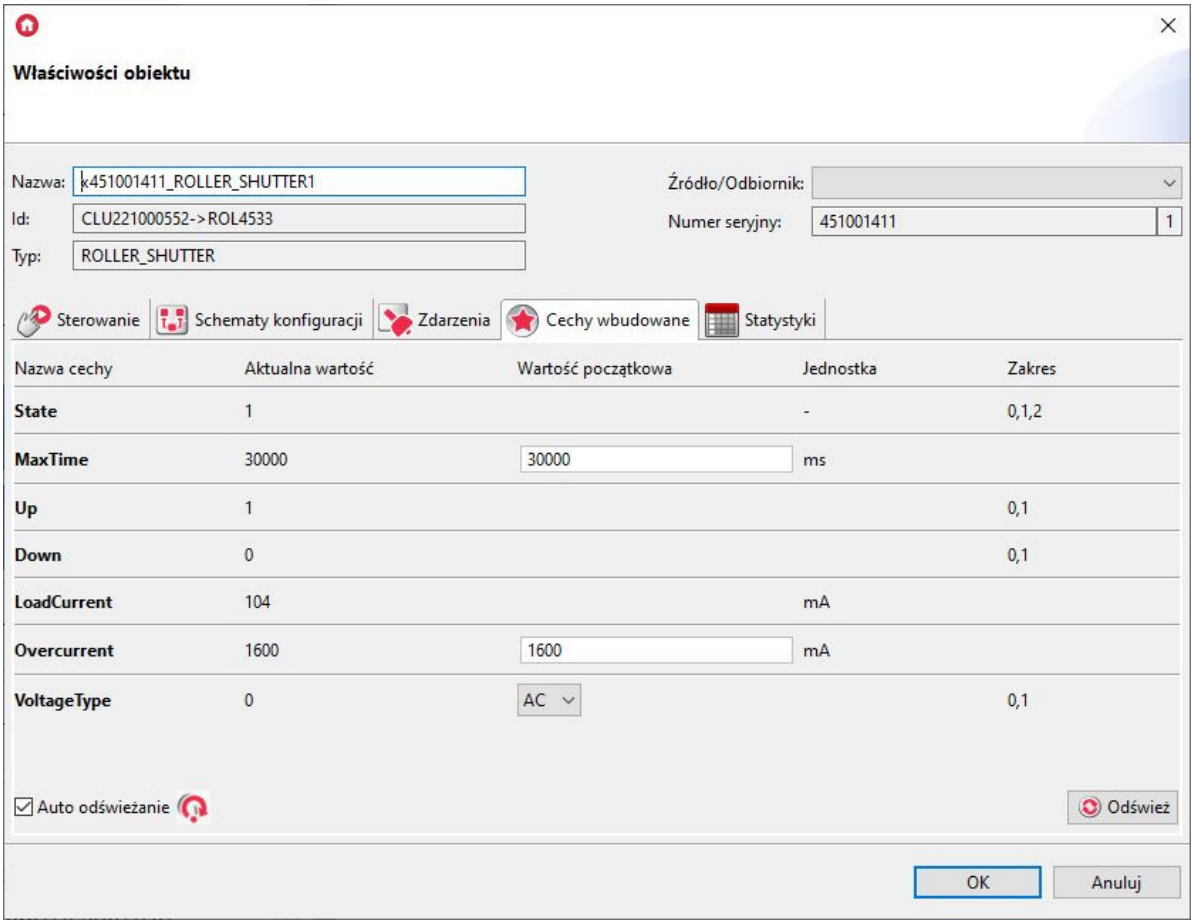
Pomiar mediów jest rejestrowany w czasie rzeczywistym - do odczytu służą cechy: `Power`, wyrażonej w watach w przypadku modułów wyjść przekaźnikowych oraz `LoadCurrent`, wyrażonej w miliamperach dla modułów sterujących napędami żaluzjowymi.

Cecha `VoltageType` przyjmuje wartości:

- Dla obiektów DOUT: 0 - AC, 1 - DC, 2 - Signal



- Dla obiektów ROLLER SHUTTER: 0 - AC, 1 - DC



XI. Funkcje serwisowe CLU

1. Przywracanie ustawień fabrycznych CLU - *Hard Reset*

Uruchomienie funkcji *Hard Reset CLU* powoduje:

- Usunięcie zapisanej konfiguracji;
- Sformatowanie partycji pamięci flash;
- Usunięcie wszystkich utworzonych obiektów LUA;
- Wyczyszczenie wszystkich danych kontrolera Z-Wave;
- Usunięcie informacji o podłączonych modułach Z-Wave.

W celu przywrócenia ustawień fabrycznych CLU funkcją *Hard Reset* należy wykonać następujące czynności (zgodnie z podaną kolejnością):

- Odłączyć zasilanie od modułu CLU;
- Nacisnąć i przytrzymać przycisk *Link* na module;
- Podłączyć zasilanie do modułu CLU;
- Trzymać wciśnięty przycisk *Link* przez co najmniej 10 sekund – obydwie diody na CLU będą świecić światłem ciągłym;
- Po upływie 10 sekund zwolnić przycisk *Link* – prawidłowe wykonanie resetu zostanie potwierdzone 5-krotnym mrugnięcie obydwu diod.

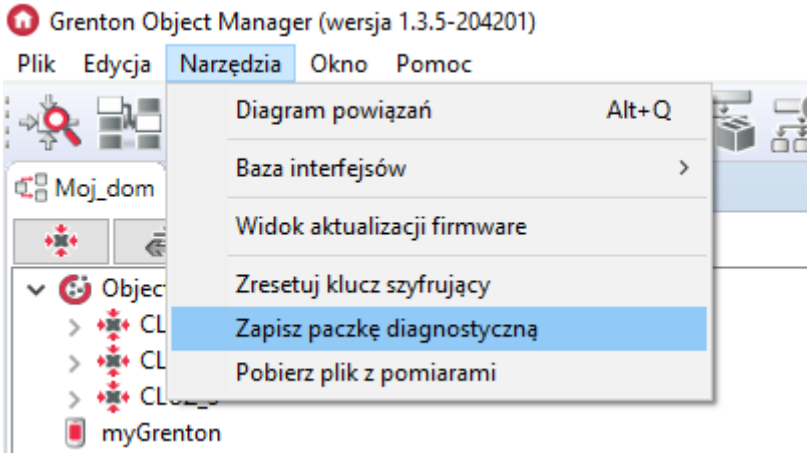
> **Uwaga!** Jeśli przed uruchomieniem funkcji *Hard Reset* do modułu CLU były dodane moduły Z-Wave, to po wykonaniu resetu niezbędne będzie wykonanie procedury usuwania i ponownego dodawania każdego modułu Z-Wave!

2. Diagnostyka systemu - *Zapisz paczkę diagnostyczną*

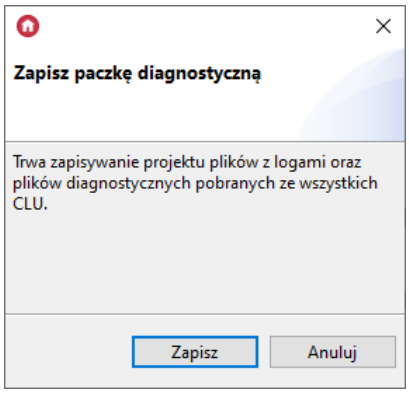
Paczka diagnostyczna wykorzystywana jest do diagnostyki jednostki centralnej CLU oraz do szybkiego odnajdywania ewentualnych problemów w stworzonym projekcie.

W celu przeprowadzenia diagnostyki systemu należy:

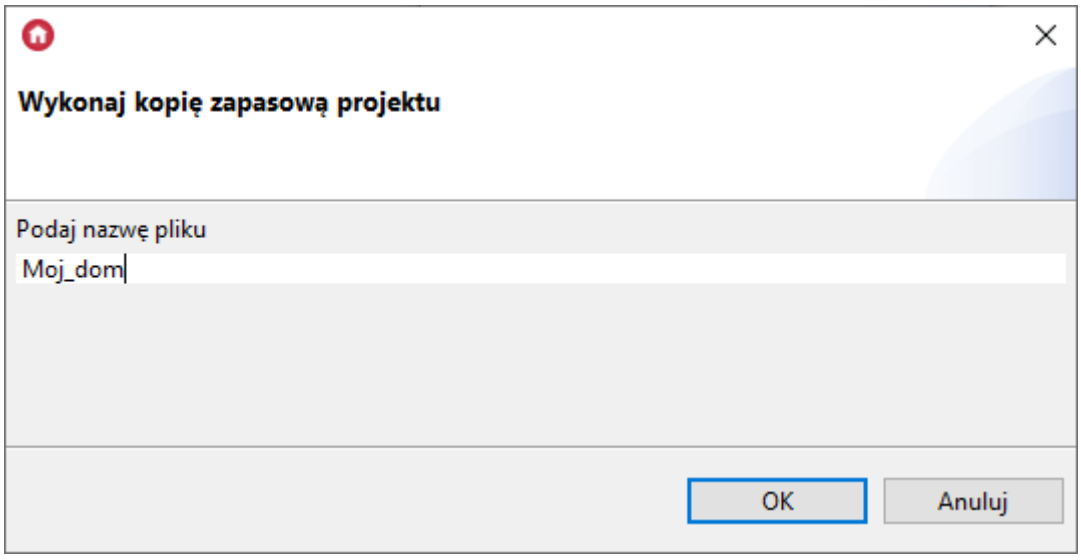
- Otworzyć projekt w Object Managerze;
- Z paska zadań wybrać `Narzędzia`, a następnie *Zapisz paczkę diagnostyczną*:



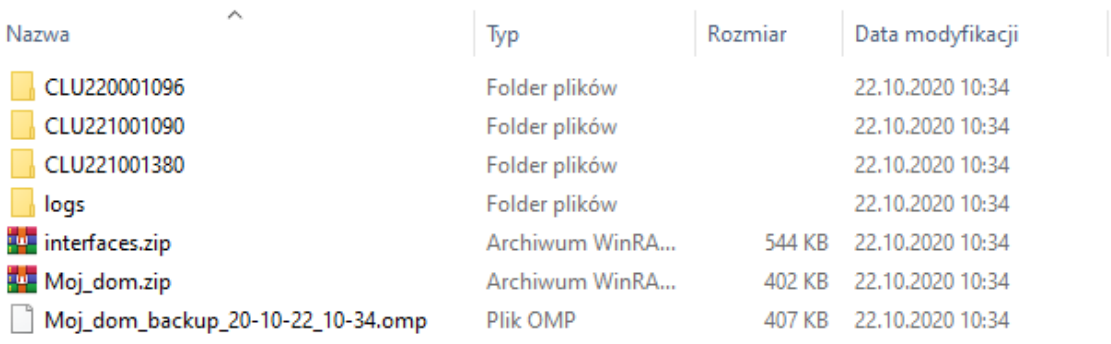
- W otwartym oknie wybrać opcję *Zapisz*:



- Określić miejsce zapisu paczki plików oraz nadać nazwę kopii zapasowej:



- Następnie w wybranej lokalizacji pojawi się paczka w formacie *.zip*, której zawartość będzie prezentowała się następująco:



- Utworzona w ten sposób paczka zawiera:
    - foldery z plikami konfiguracyjnymi wszystkich CLU;
    - folder "logs" zawierający plik z wyszczególnionymi logami aplikacji;
    - paczkę .zip zawierającą bazę interfejsów wykorzystaną w projekcie;
    - paczkę .zip zawierającą informacje na temat projektu;
    - plik kopii zapasowej projektu.

XII. SMART PANEL

1. Wyposażenie Smart Panelu

Smart Panel składa się z:

- Wyświetlacza OLED;
- Czterech przycisków dotykowych;
- Sensora rozpoznającego cztery gesty;
- Sensora zbliżenia/obecności;
- Sensora temperatury;
- Sensora natężenia światła;
- Buzzera.

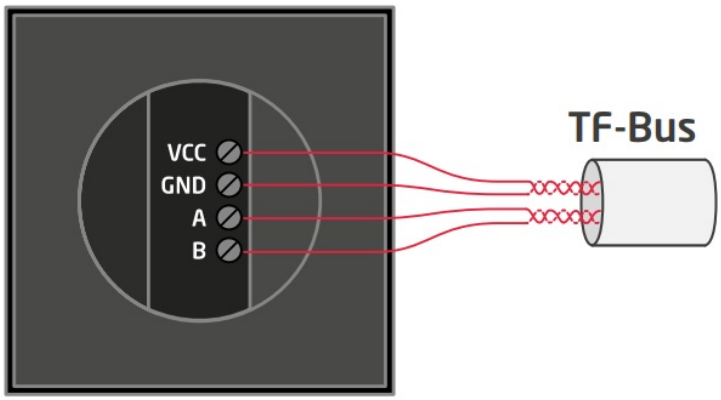
2. Podłączenie Smart Panelu do CLU

> **Uwaga!** Smart Panel dostępny jest dla Object Managera w wersji 1.2.0.180202 i wyższej oraz dla CLU z firmware 04.07.29-1802 i wyższym.

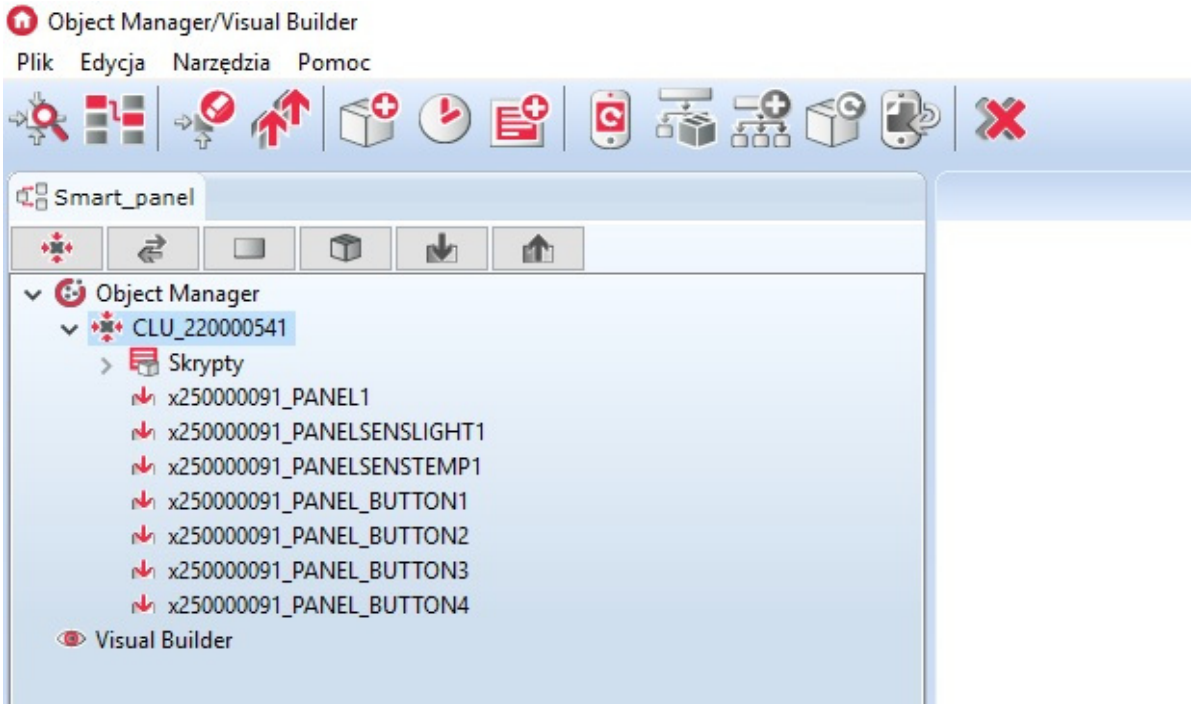
> **Uwaga!** Smart Panel w wersji v4 dostępny jest dla Object Managera w wersji 1.2.1.190201 i wyższej oraz dla CLU z firmware 04.07.49-1912 i wyższym.

Podłączenie Smart Panelu do systemu odbywa się przy pomocy skrętki. Do odpowiednich zacisków złącza ARK należy wyprowadzić od Smart Panelu dwie pary skręconych przewodów - schemat podłączenia przedstawia poniższy rysunek:

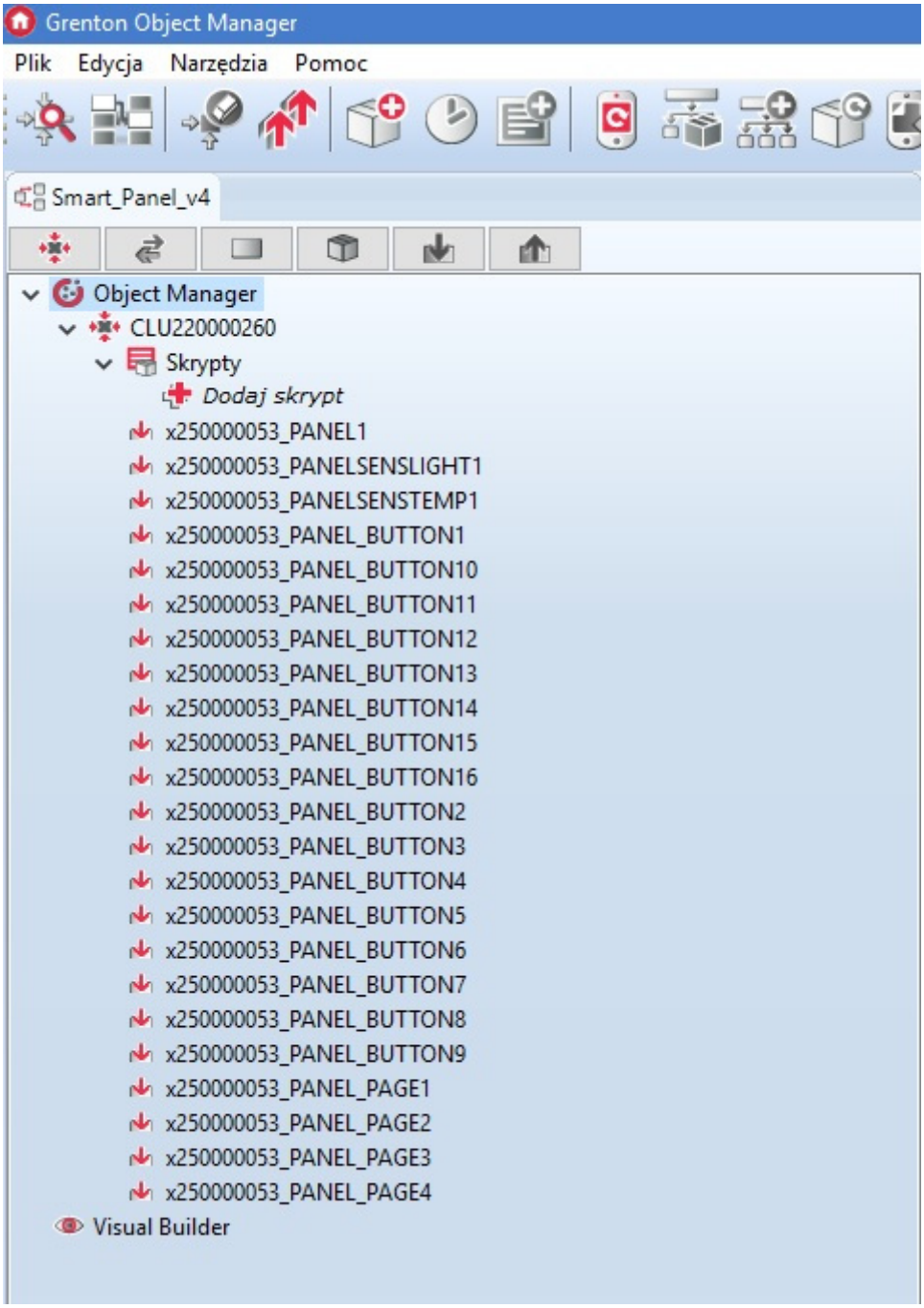
- Jeden przewód z pierwszej skręconej pary (np. kabla UTP) podłączyć do zacisku Vcc;
- Drugi przewód z pary podłączyć do zacisku GND;
- Po jednym przewodzie z drugiej pary podłączyć do zacisków A i B.



Po podłączeniu i przeprowadzeniu w projekcie operacji *CLU Discovery* na liście modułów pojawią się następujące elementy Smart Panelu v3:



Po podłączeniu i przeprowadzeniu w projekcie operacji *CLU Discovery* na liście modułów pojawią się następujące elementy Smart Panelu dla wersji v4:



W przypadku poprawnego dodania elementów do projektu, można przejść do tworzenia konfiguracji.

> **Uwaga!** W przypadku niepowodzenia należy skontaktować się z działem Support!

3. Informacje pomocne przy tworzeniu konfiguracji

1. Konfiguracja panelu z wyświetlaczem różni się od konfiguracji klasycznego panelu dotykowego Grenton między innymi tym, że oprócz: cech, metod oraz zdarzeń każdego przycisku, czujnika temperatury/natężenia światła, użytkownik ma również do dyspozycji: sensor gestów, a także cechy, metody i zdarzenia dla samego *Smart Panelu*.

   Od wersji 04.03.04.1910 dostępne są nowe funkcjonalności *Smart Panelu*, takie jak: obiekt konfiguracyjny strony *PANEL_PAGE* czy nowe cechy, metody i zdarzenia w obiekcie *PANEL*.

2. Wyświetlacz, w który wyposażony jest panel dotykowy, ma rozdzielczość 128x64 pikseli.

3. Smart Panel v3 może pracować w dwóch trybach pracy: wyświetlającym ikony (wyświetlacz podzielony jest na 4 pola) lub w trybie rysowania (wykorzystując całe pole wyświetlacza).

   Smart Panel v4 może pracować w czterech trybach pracy:

   1. Tryb kompatybilności wstecz (domyślna konfiguracja) - `Inactive`,
   2. Tryb wyświetlania ikon (wyświetlacz podzielony na 4 pole) - `Buttons`,
   3. Tryb rysowania (wykorzystując całe pole wyświetlacza) - `FreeDraw`,
   4. Tryb obsługi termostatów - `Thermostats`.
4. Panel dotykowy wyposażony jest w slot karty microSD, która służy do przechowywania domyślnych ikon wyświetlanych na panelu. Pliki muszą być umieszczone w głównym katalogu karty z rozszerzeniem *.bmp*.
5. Ekran Smart Panelu domyślnie jest wygaszony. Zapala się w chwili, gdy zadziała czujnik zbliżenia (czas wyświetlania jest pobierany z cechy *Panel->ProximityTimeout* - po tym czasie, gdy panel nie wykrywa obecności, wyświetlacz wygasza się).
6. Czujnik obecności działa w zależności od odległości ustawianej przy pomocy czułości - cecha `ProximitySens`. Po wykryciu obecności generowane jest zdarzenie `OnProximityDetect`.

4. Konfiguracja modułu Smart Panel w wersji v3

4.1. Parametry konfiguracyjne

A. Panel

CECHY

Nazwa	Opis
`GestureIconUp`	Nazwa pliku BMP z ikoną dla gestu w górę (bez rozszerzenia)
`GestureIconDown`	Nazwa pliku BMP z ikoną dla gestu w dół (bez rozszerzenia)
`GestureIconLeft`	Nazwa pliku BMP z ikoną dla gestu w lewo (bez rozszerzenia)
`GestureIconRight`	Nazwa pliku BMP z ikoną dla gestu w prawo (bez rozszerzenia)
`ProximitySens`	Czułość sensora zbliżeniowego
`ProximityTimeout`	Czas, po którym wyświetlacz zostanie wygaszony
`ProximityValue`	Sygnał sensora zbliżeniowego (wartość bezwymiarowa)
`BuzzerValue`	Sterowanie sygnalizacją dźwiękową (załącz/wyłącz)

METODY

Nazwa	Opis
`SwitchOnDisplay`	Wybudza wyświetlacz z trybu uśpienia
`ShowButtons`	Zmienia tryb wyświetlacza na *buttons*
`ClearScreen`	Czyści zawartość wyświetlacza w trybie *freedraw*
`PrintText`	Wyświetla tekst w trybie *freedraw*
`PrintFloat`	Wyświetla liczbę w trybie *freedraw*
`DrawLine`	Rysuje linię w trybie *freedraw*
`DrawPoint`	Rysuje punkt w trybie *freedraw*
`DrawIcon`	Rysuje ikonę (bmp) w trybie *freedraw*
`DisplayContent`	Wyświetla zawartość bufora pamięci graficznej; zmienia tryb wyświetlacza na *freedraw*
`SetGestureIconUp`	Ustawia plik BMP z ikoną dla gestu w górę
`SetGestureIconDown`	Ustawia plik BMP z ikoną dla gestu w dół
`SetGestureIconLeft`	Ustawia plik BMP z ikoną dla gestu w lewo
`SetGestureIconRight`	Ustawia plik BMP z ikoną dla gestu w prawo
`SetProximitySens`	Ustawia czułość sensora zbliżeniowego
`SetProximityTimeout`	Ustawia czas, po którym wyświetlacz zostanie wygaszony
`SetBuzzerValue`	Załącza/wyłącza sygnalizację dźwiękową

ZDARZENIA

Nazwa	Opis
`OnGestureUp`	Zdarzenie związane z gestem w górę
`OnGestureDown`	Zdarzenie związane z gestem w dół
`OnGestureLeft`	Zdarzenie związane z gestem w lewo
`OnGestureRight`	Zdarzenie związane z gestem w prawo
`OnProximityDetect`	Zdarzenie wywołane w przypadku wykrycia osoby zbliżającej się do wyświetlacza panelu

B. Przyciski

CECHY

Nazwa	Opis
`Mode`	Zwraca ustawiony tryb działania przycisku: 0 - monostabilny, 1 - bistabilny, 2 - zablokowany (dioda świeci na czerwono światłem ciągłym)
`HoldDelay`	Czas (w milisekundach), po jakim wyzwolone zostanie zdarzenie `OnHold` (przy wciśnięciu i trzymaniu przycisku)
`HoldInterval`	Cykliczny odstęp czasu (w milisekundach), po jakim podczas trzymania przycisku wyzwalane będzie zdarzenie `OnHold`
`Value`	Zwraca stan wejścia (0 lub 1)
`Label`	Tekst opisujący przycisk (wyświetlany zamiast ikony)
`IconA`	Nazwa pliku ikony przypisanej do przycisku w trybie monostabilnym oraz bistabilnym w pozycji *OFF*; nazwa poprzedzona „~" wyświetli grafikę w negatywie; `IconA` ma priorytet nad cechą `Label`
`IconB`	Nazwa pliku ikony przypisanej do przycisku w trybie bistabilnym w pozycji *ON*; nazwa poprzedzona „~" wyświetli grafikę w negatywie

METODY

Nazwa	Opis
`SetMode`	Ustawia tryb działania przycisku: 0 - monostabilny, 1 - bistabilny, 2 - zablokowany (dioda świeci na czerwono światłem ciągłym)
`SetHoldDelay`	Ustawia wartość `HoldDelay`
`SetHoldInterval`	Ustawia wartość `HoldInterval`
`SetLabel`	Ustawia tekst opisujący przycisk
`SetIconA`	Ustawia plik ikony A
`SetIconB`	Ustawia plik ikony B
`ShowOK`	Powoduje miganie zielonej diody na przycisku przez dwie sekundy (częstotliwość 500ms)
`ShowError`	Powoduje miganie czerwonej diody na przycisku przez dwie sekundy (częstotliwość 500ms)
`LedSwitchOn`	Załącza zieloną diodę na przycisku
`LedSwitchOff`	Wyłącza zieloną diodę na przycisku

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane przy zmianie stanu (niezależnie od wartości)
`OnSwitchOn`	Zdarzenie wywoływane w momencie ustawienia stanu wysokiego na wejściu
`OnSwitchOff`	Zdarzenie wywoływane w momencie ustawienia stanu niskiego na wejściu
`OnShortPress`	Zdarzenie wywoływane po naciśnięciu przycisku na okres 500 - 2000ms
`OnLongPress`	Zdarzenie wywoływane po naciśnięciu przycisku na okres 2000 – 5000ms
`OnHold`	Zdarzenie wywoływane po raz pierwszy po upłynięciu czasu `HoldDelay`, a następnie cyklicznie co czas `HoldInterval`
`OnClick`	Zdarzenie wywoływane po naciśnięciu przycisku na czas poniżej 500ms

C. Czujniki temperatury i oświetlenia

CECHY

Nazwa	Opis
`Threshold`	Wielkość histerezy (dokładność 0,1) określająca czułość, przy której następuje wygenerowanie zdarzeń: `OnChange`, `OnLowerValue`, `OnRaiseValue`
`Sensitivity`	Czas (w ms), dla którego próbkowane wartości są uśredniane
`MinValue`	Minimalna wartość cechy `Value`, której przekroczenie wywołuje zdarzenie `OnOutOfRange`
`MaxValue`	Maksymalna wartość cechy `Value`, której przekroczenie wywołuje zdarzenie `OnOutOfRange`
`Value`	Wartość wejścia: dla czujnika temperatury (od 0 do 45°C) lub dla czujnika światła (0 – 100%)

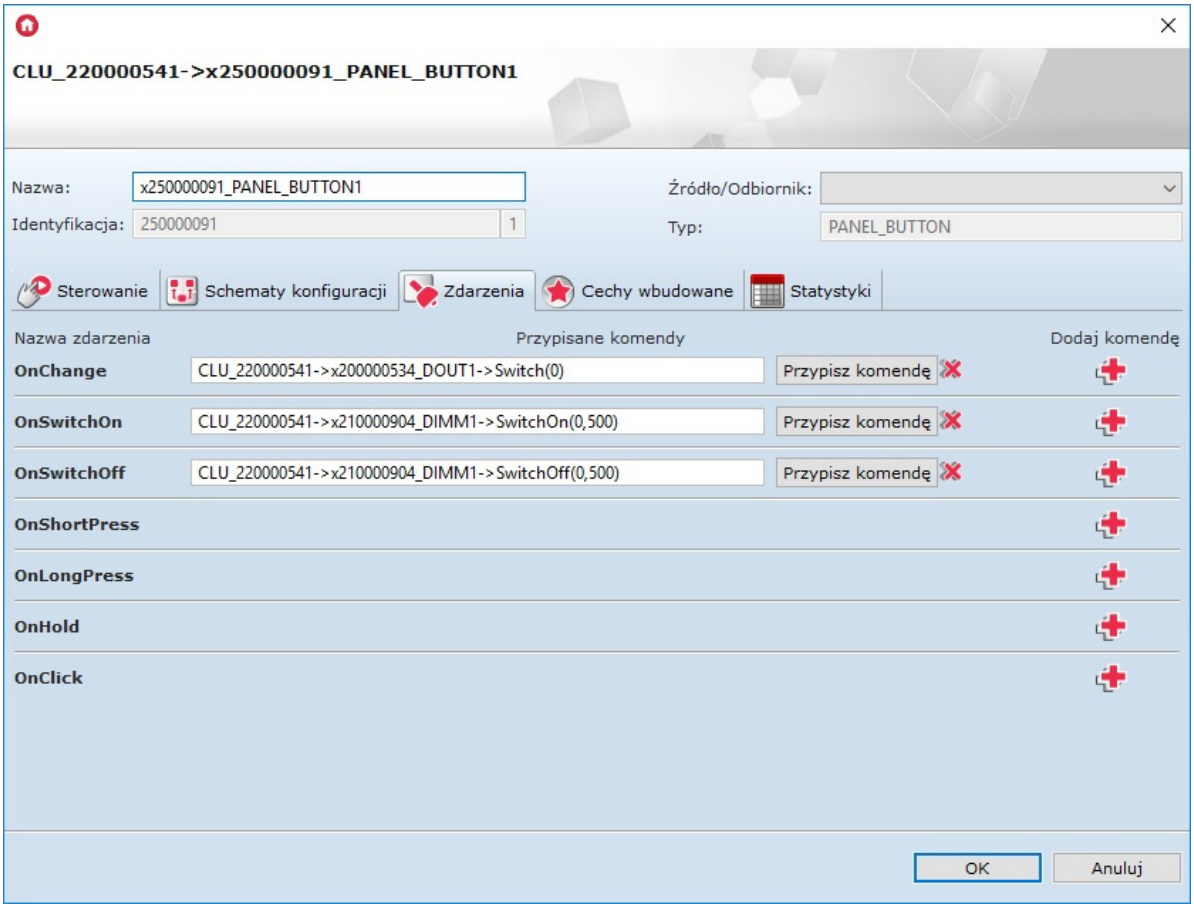
ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w momencie zmiany stanu wejścia (niezależnie od wartości)
`OnRaiseValue`	Zdarzenie wywoływane przy przekroczeniu górnego progu histerezy
`OnLowerValue`	Zdarzenie wywoływane przy przekroczeniu dolnego progu histerezy
`OnOutOfRange`	Zdarzenie wywoływane, gdy wartość na wyjściu znajduje się poza wyznaczonym zakresem

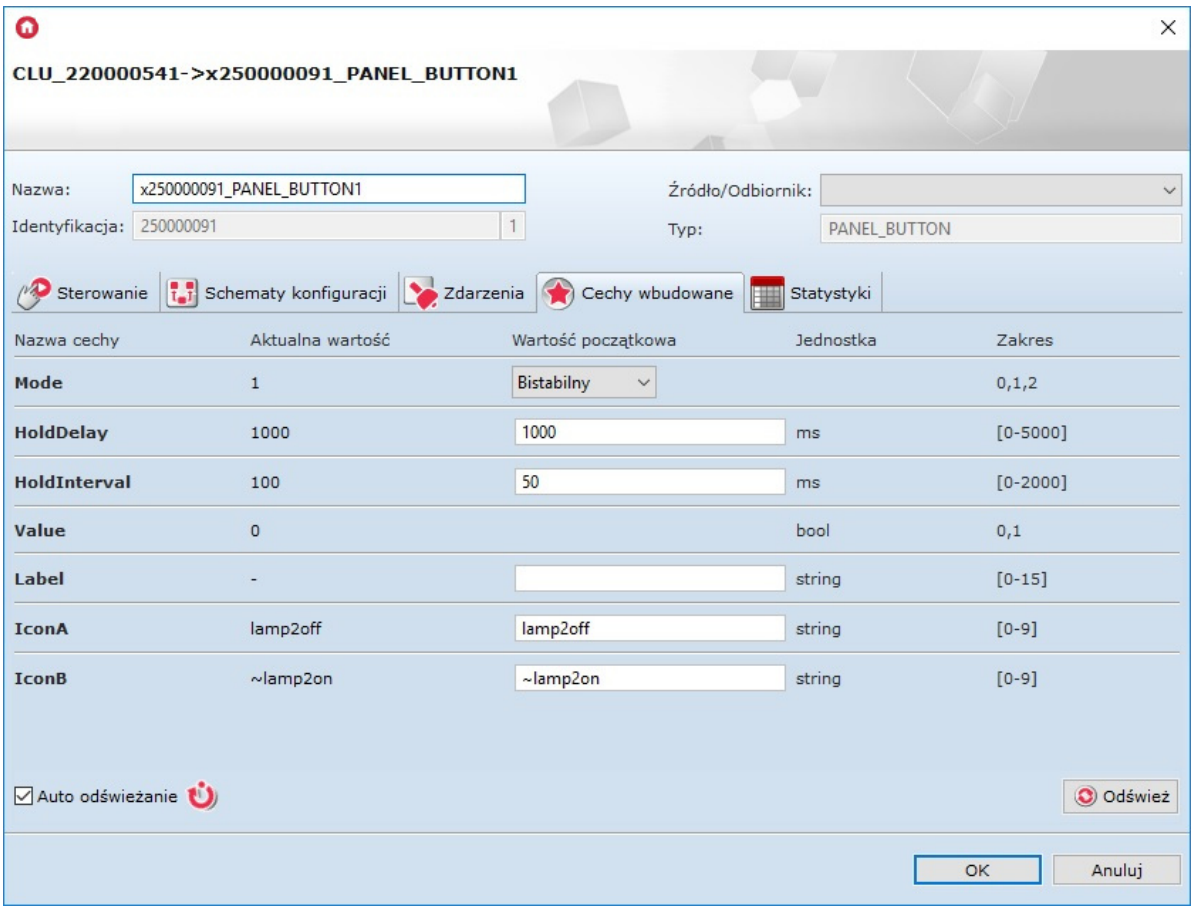
4.2. Tworzenie konfiguracji przycisków i wyświetlacza

W celu stworzenia konfiguracji należy:

- Otworzyć obiekt *PANEL_BUTTONX* (gdzie X to numer jednego z 4 przycisków) poprzez dwukrotne kliknięcie na liście modułów;
- Przejść do zakładki *Zdarzenia*;
- Skonfigurować działanie przycisku poprzez przypisanie metod do określonych zdarzeń (klikając w „+" po prawej stronie okna):



- Wybrać zakładkę *Cechy wbudowane* i zdefiniować obiekty wyświetlane na ekranie danego przycisku:
    - `Label` - cecha określająca tekst przypisany do danego przycisku;
    - `IconA` - cecha określająca nazwę ikony przypisanej do danego przycisku, gdy znajduje się w trybie monostabilnym, bądź dla trybu bistabilnego dla cechy `Value` =0;
    - `IconB` - cecha określająca nazwę ikony przypisanej do danego przycisku, gdy znajduje się w trybie bistabilnym dla cechy `Value` =1. Chcąc przypisać tę samą ikonę, ale o odwróconej kolorystyce należy nazwę piktogramu poprzedzić znakiem „~" (np. `~lamplon`):



Powyższe cechy można ustawić zarówno w zakładce *Cechy wbudowane*, jak również za pośrednictwem metod: `SetLabel`, `SetIconA`, `SetIconB`.

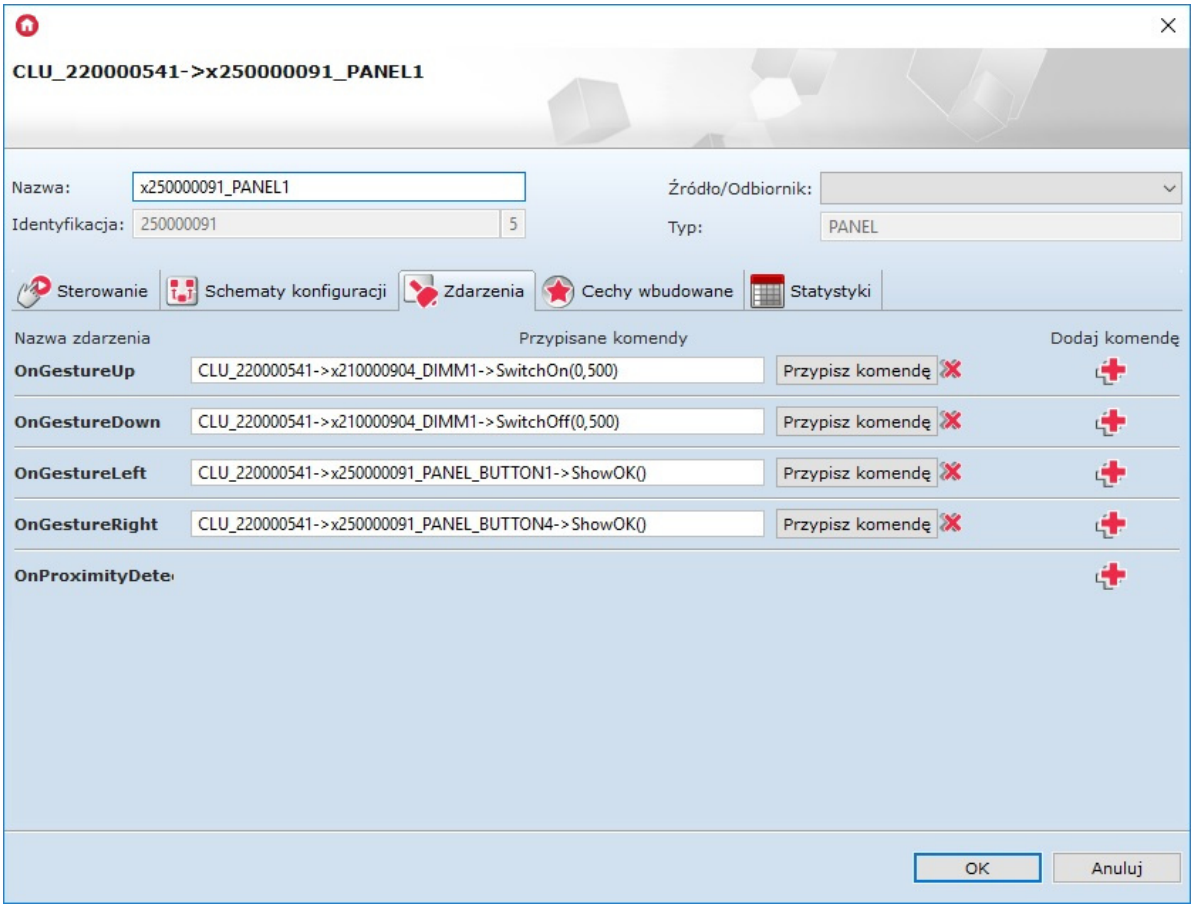
> **Uwaga!** Metoda `SetIcon` ma w systemie wyższy priorytet niż metoda `SetLabel`!

- Wysłać konfigurację do CLU.

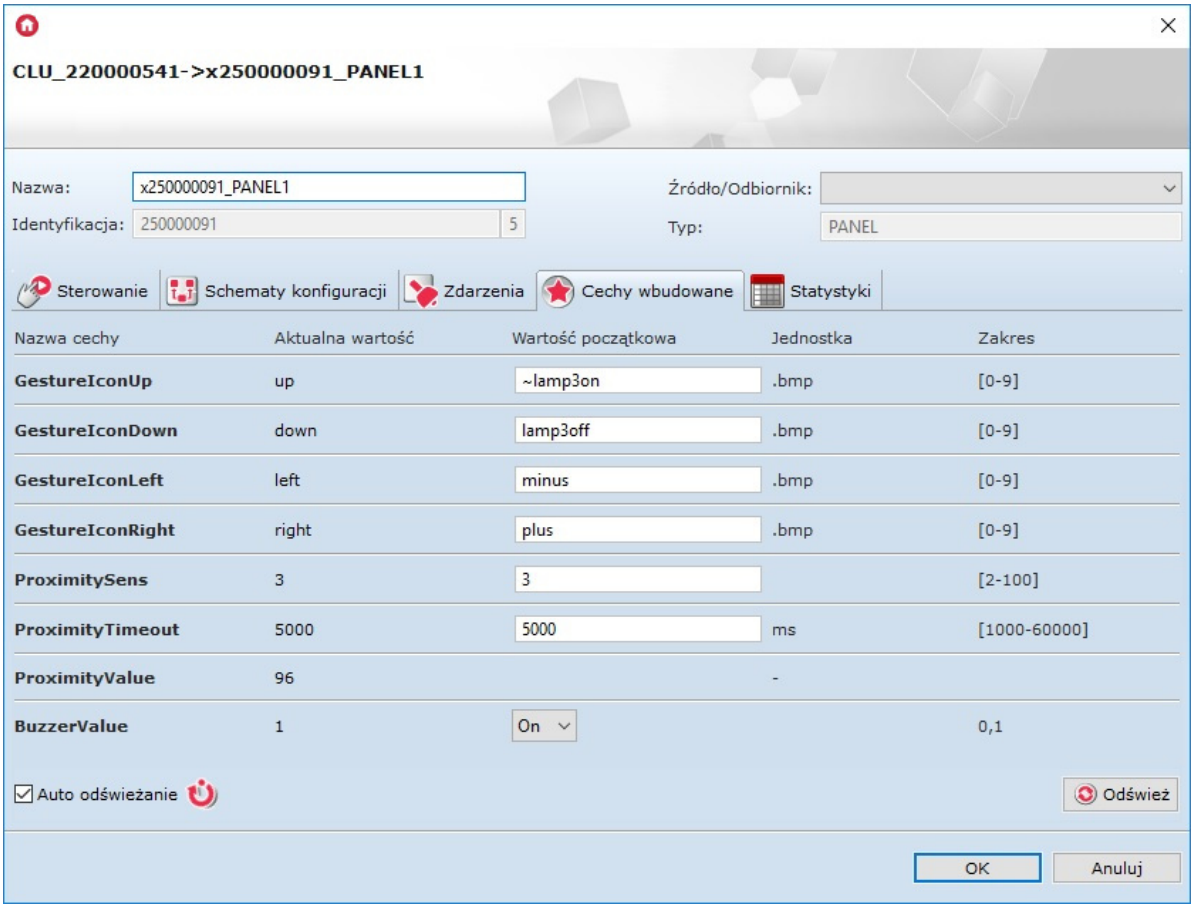
4.3. Tworzenie konfiguracji sensora gestów

Chcąc utworzyć konfigurację dla sensora gestów należy:

- Otworzyć – przez dwuklik - obiekt *Panel*;
- Przejść do zakładki *Zdarzenia*;
- Przypisać metody do zdarzeń `OnGestureUp`, `OnGestureDown`, `OnGestureLeft`, `OnGestureRight` (klikając w „+" po prawej stronie każdej z metod):



Możliwa jest podmiana domyślnie wyświetlanych ikon przy wywoływaniu gestów – w tym celu należy przejść do zakładki *Cechy wbudowane* i wpisać nazwy żądanych ikon bez rozszerzenia *.bmp*:



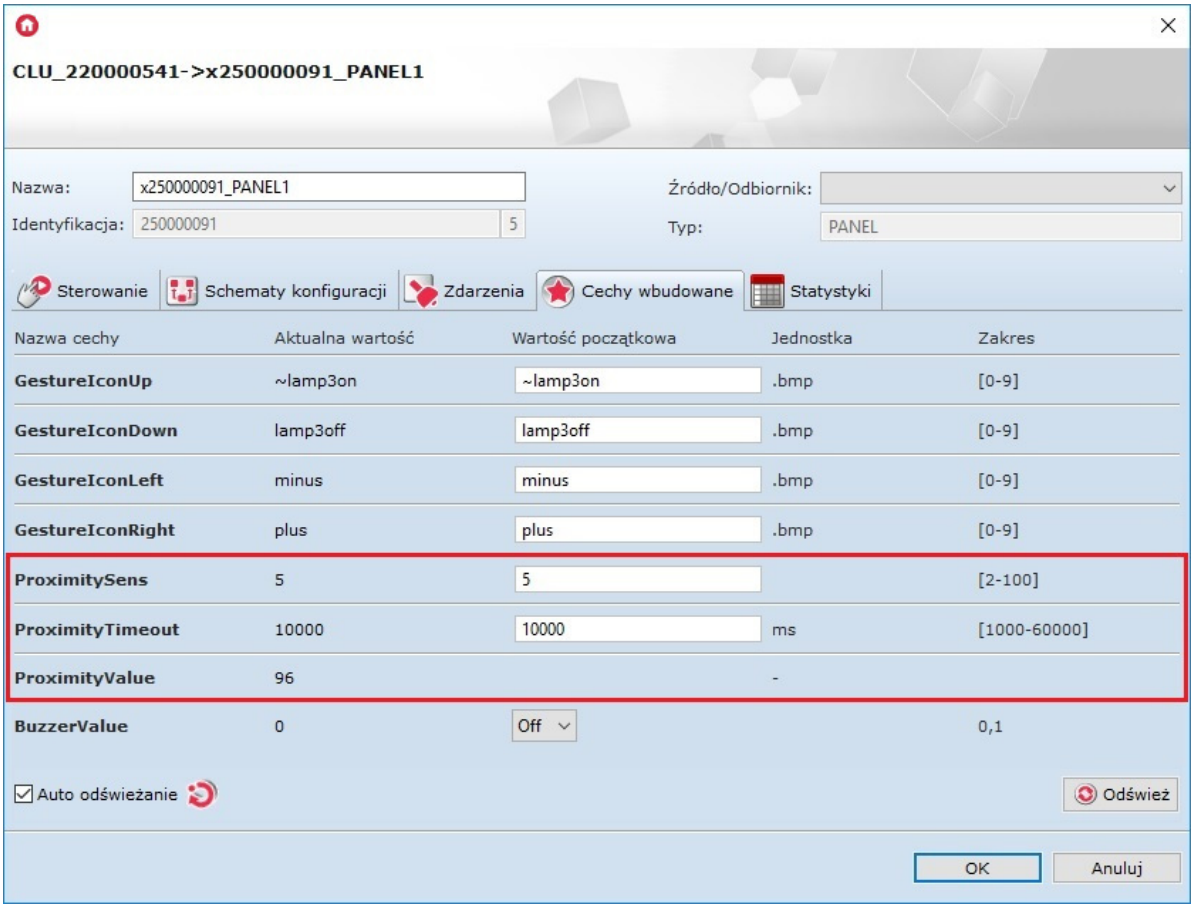
Użycie ikon będzie możliwe, gdy będą wgrane na kartę microSD z rozszerzeniem *.bmp*.

- Zatwierdzić okno konfiguracji przyciskiem *OK*;
- Wysłać konfigurację do CLU.

4.4. Konfiguracja czujnika zbliżenia

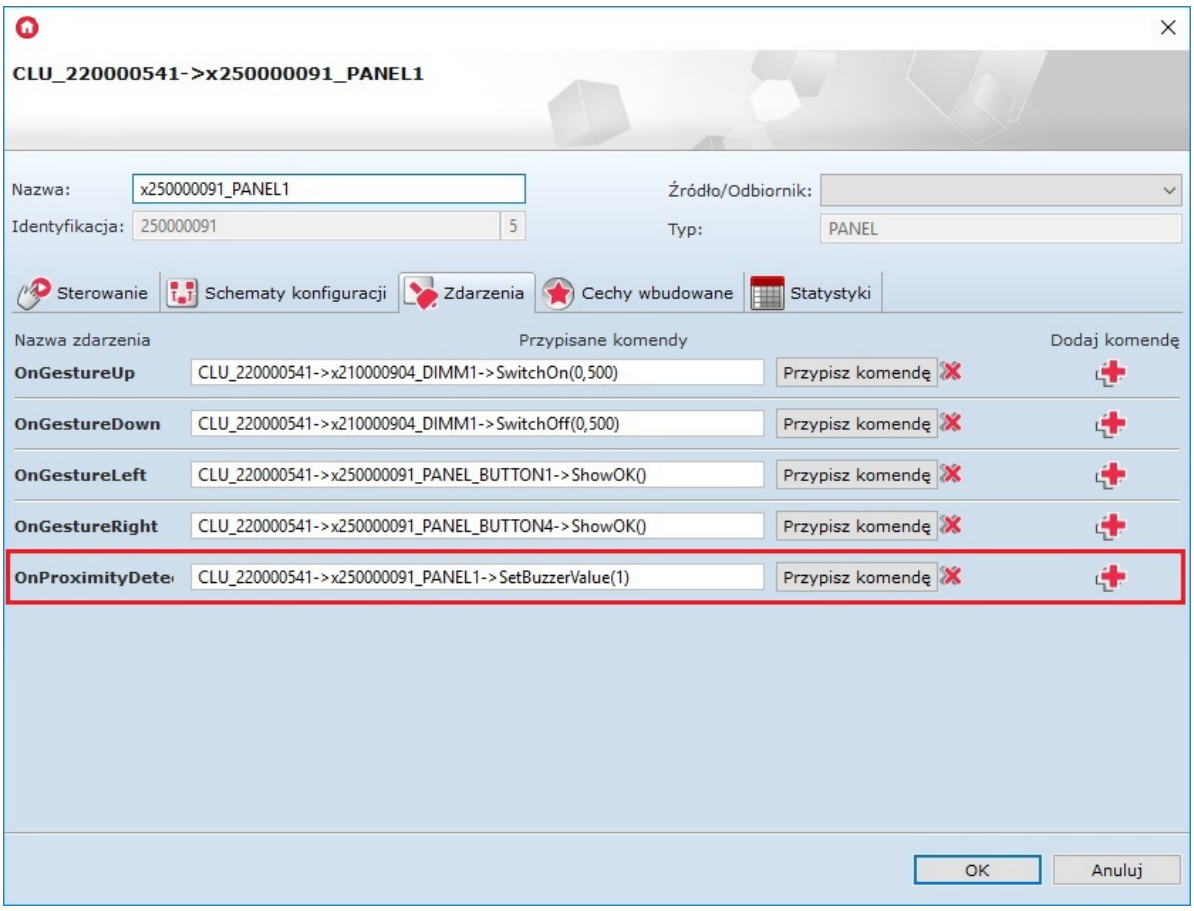
W celu ustawienia parametrów czujnika zbliżenia należy:

- Otworzyć – przez dwuklik - obiekt *Panel*;
- Przejść do zakładki *Cechy wbudowane*, gdzie znajdują się 3 cechy odnoszące się do czujnika zbliżenia:
  - `ProximitySens` - określa czułość sensora;
  - `ProximityTimeout` – definiuje czas, po jakim wyświetlacz zostaje wygaszony, gdy nie zostanie wykryty ruch;
  - `ProximityValue` – zwraca przybliżoną odległość w centymetrach od panelu do obiektu:



Powyższe cechy można ustawiać zarówno w zakładce *Cechy wbudowane*, jak również przy pomocy metod: `SetProximitySens` oraz `SetProximityTimeout` (w metodach obiektu *Panel*).

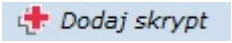
- Reakcja czujnika zbliżenia generuje zdarzenie `OnProximityDetect`, do którego można dodać dodatkowe metody:



- Wysłać konfigurację do CLU.

4.5. Tworzenie konfiguracji wielu stron panelu dotykowego

Chcąc rozpocząć tworzenie konfiguracji panelu obsługującego wiele stron, na CLU należy utworzyć cechę typu *number* (określa numer strony startowej) o przykładowej nazwie *strona* - w tym celu kliknąć dwukrotnie na CLU, przejść do zakładki *Cechy użytkownika* i wybrać przycisk: 

Aby panel mógł wyświetlać żądaną zawartość na ekranie, konieczne jest utworzenie skryptu (np. *Wyswietlacz*) z kilkoma stronami – w tym celu wybrać przycisk przy lewej krawędzi okna programu Object Manager: 

> **Uwaga!** Nazwa skryptu nie może zawierać polskich znaków!

- **STRONA Z PRZYCISKAMI** - Do skryptu należy dodać warunek sprawdzający, jaki jest aktualny numer strony (wartość *Cechy użytkownika*: *strona*), a dla spełnionego warunku - dla konkretnej strony - dodać akcję przypisania ikon wszystkich 4 przycisków (metody `SetIconA` dla elementów *PANEL_BUTTON1-4*) oraz metodę `ShowButtons` wyświetlającą na ekranie panelu wybrane ikony;

> **Uwaga!** Poza przypisaniem ikon do określonych przycisków wymagane jest wywołanie metody `ShowButtons`, gdyż samo ich przypisanie nie spowoduje, że pojawią się na wyświetlaczu!

> **Uwaga!** W przypadku tworzenia wielu stron, ustawienie przycisku w tryb bistabilny - przy pomocy cechy/metody - nie będzie poprawnie odczytywało stanu przekaźnika (z uwagi na różne funkcjonalności przycisków przy zmianach stron)!!

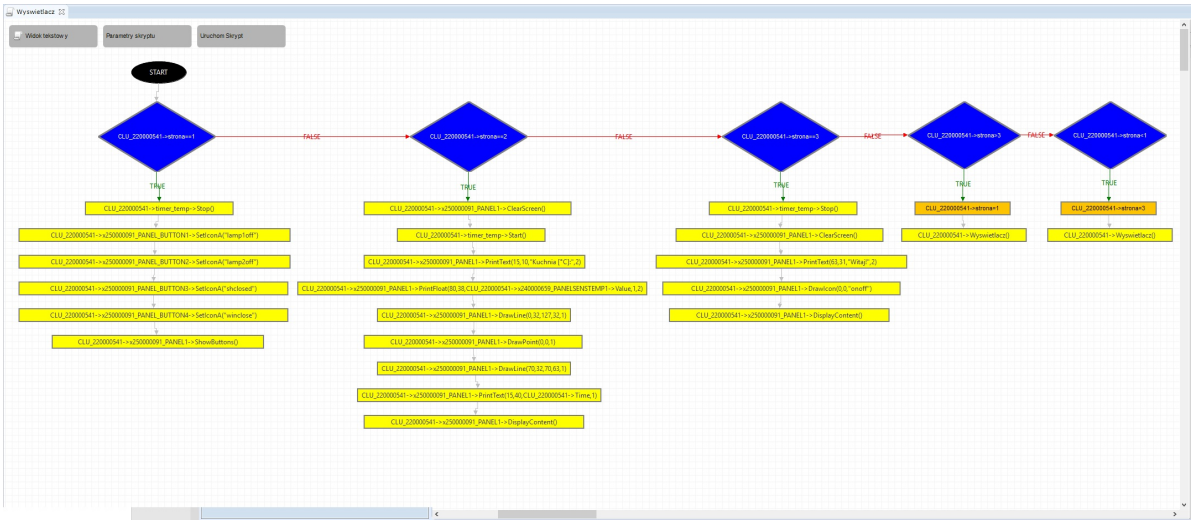
- **STRONA Z GRAFIKAMI I TEKSTAMI** – Przy projektowaniu strony zawierającej grafiki oraz teksty należy dodać:
  - warunek sprawdzający numer strony (nie może być to strona z przyciskami);
  - akcję *PANEL->*`ClearScreen()`;
  - akcje ustawienia tekstu oraz linii (opisane poniżej);
  - akcję *PANEL->*`DisplayContent()`.

Akcje ustawiania tekstu i linii:

- *PANEL->*`PrintText` – metoda powodująca wypisanie tekstu lub cechy – do jej wywołania należy podać cztery parametry: początkowe współrzędne na ekranie (x, y), tekst oraz rozmiar czcionki (gdzie 1 – 10 pkt, 2 – 14 pkt, 3 – 28 pkt);
- *PANEL->*`PrintFloat` – metoda działająca tak samo jak `PrintText`, z tą różnicą, że posiada dodatkowy parametr *Precision*, odpowiedzialny za ilość miejsc po przecinku parametru *number*;
- *PANEL->*`DrawLine` – metoda rysująca linię – do jej wywołania konieczne jest podanie 5 parametrów: współrzędne początkowe (x, y), końcowe (xe, ye) oraz kolor linii (gdzie 0 – czarny, 1 - biały);
- *PANEL->*`DrawPoint` – metoda rysująca punkt – do jej wywołania należy podać 3 parametry: współrzędne (x, y) oraz kolor (parametr działa jak przy wywołaniu metody `DrawLine`);
- *PANEL->*`DrawIcon` – metoda rysująca ikonę – do jej wywołania trzeba podać 3 parametry: współrzędne początkowe (x, y) oraz nazwę ikony z zasobnika.

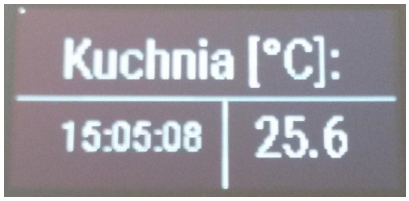
- **ZAPĘTLENIE SKRYPTU** – Do skryptu należy dodać warunki, które spowodują, że przy wygenerowaniu gestu w prawo na ostatniej stronie, panel wróci do pierwszej strony (i na odwrót) - tak, by uzyskać działanie pętli.

Realizację wszystkich powyżej opisanych metod zaprezentowano na zrzucie ekranu przykładowego skryptu:

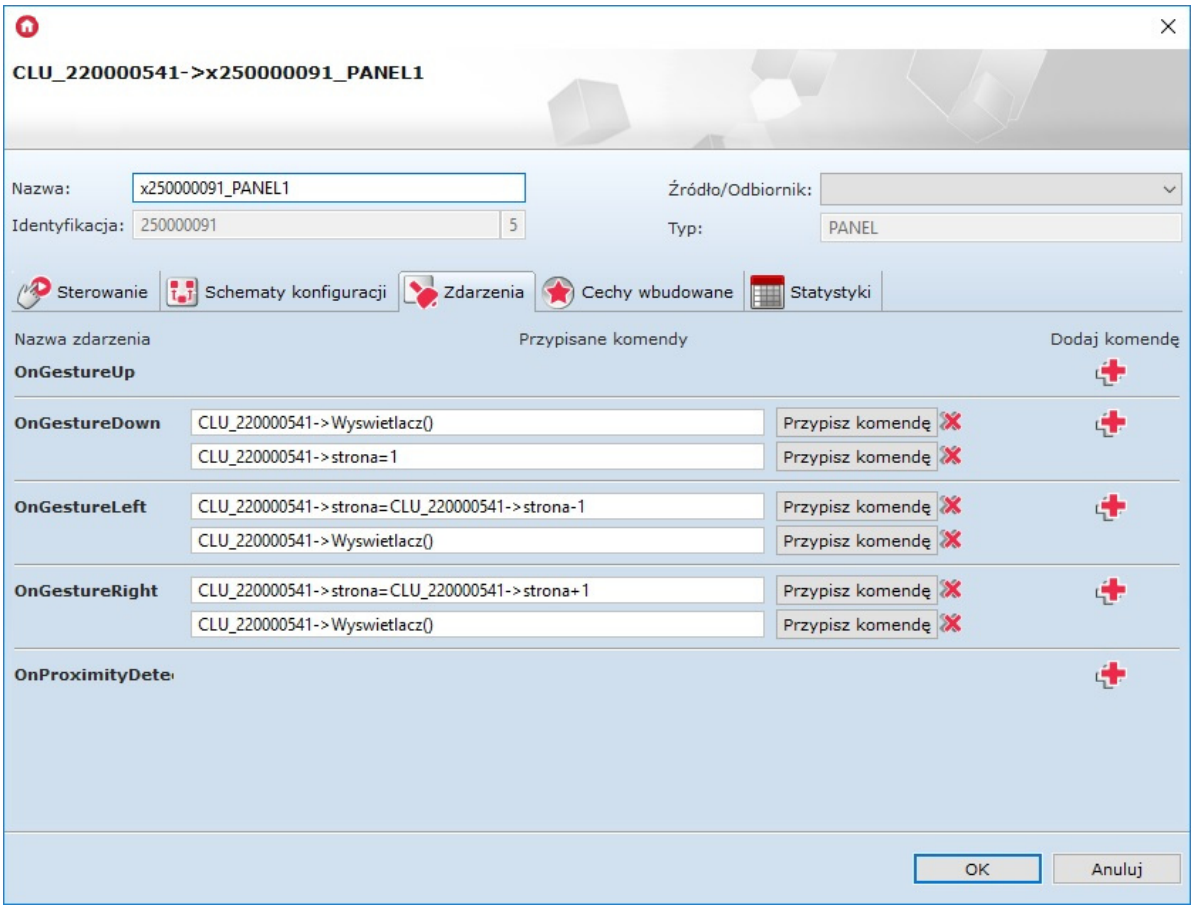


Powyższy skrypt zamieszczono na końcu dokumentu w wersji tekstowej (punkt 3.).

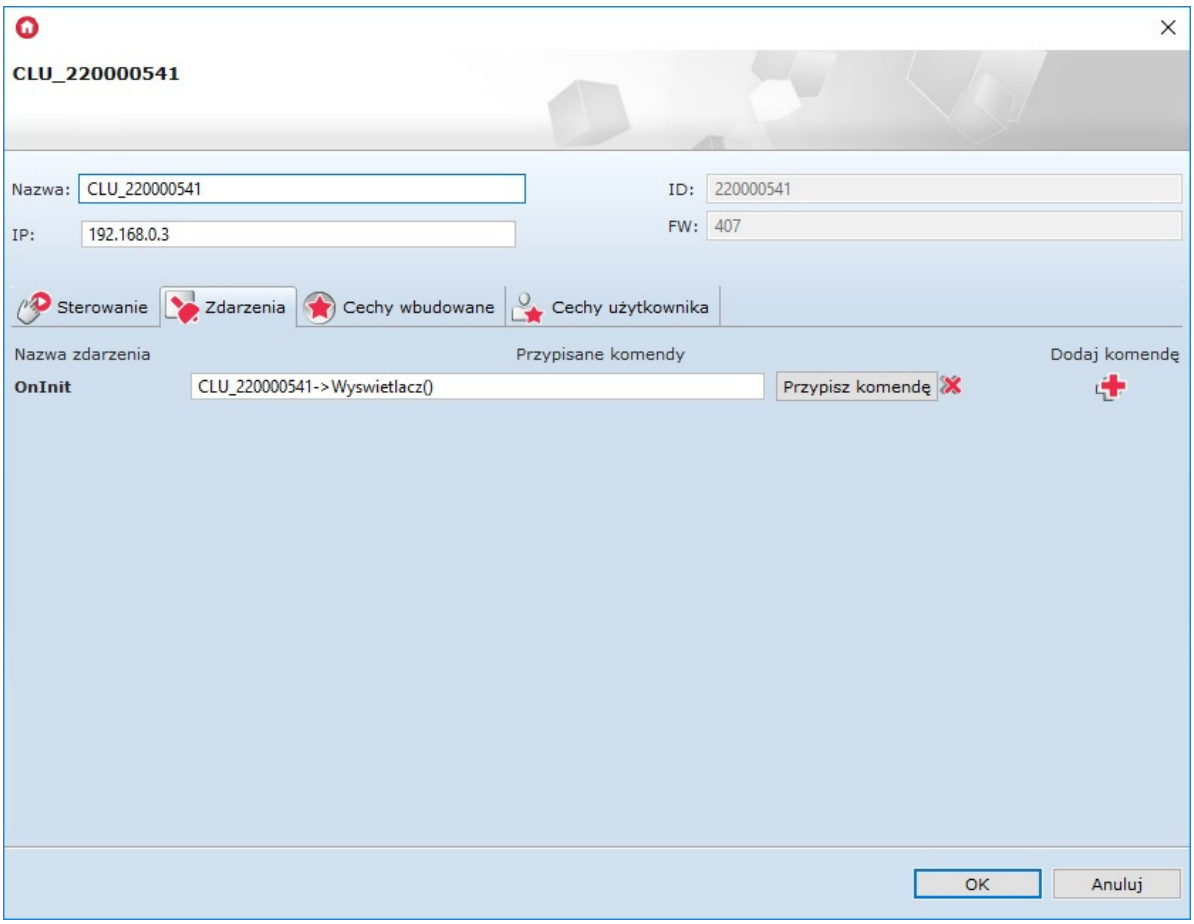
Druga strona zaprogramowana w skrypcie będzie wyglądać następująco:



- W następnym kroku - do gestów panelu w lewo i w prawo - należy przypisać operacje zwiększania zmiennej użytkownika *strona* oraz uruchamiania skryptu *Wyswietlacz* jak na rysunku poniżej:

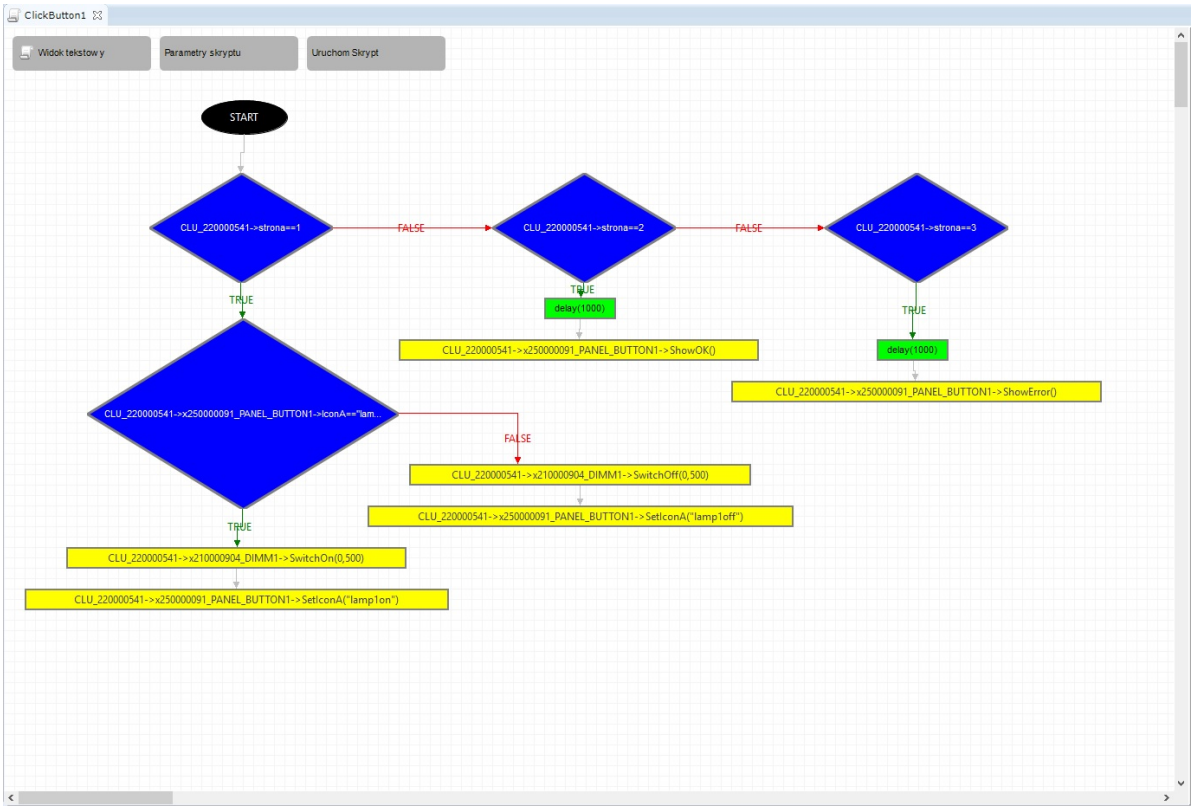


- Przypisać do zdarzenia *CLU->* `OnInit` wywołanie skryptu *Wyswietlacz*:



- Stworzyć skrypt (np. *ClickButton1*) do obsługi zdarzenia `OnClick` jednego wybranego przycisku na każdej stronie – należy utworzyć osobne skrypty dla każdego przycisku:
    - Dodać warunek sprawdzający numer strony;
    - Chcąc zrealizować funkcję trybu bistabilnego dla przycisku, dodać kolejny warunek sprawdzający aktualny stan ikony i podejmujący odpowiednie działania (załączający lub wyłączający, np. oświetlenie);
    - Dodać kolejne warunki sprawdzające numer strony.

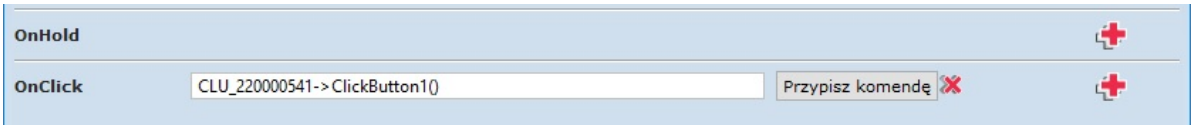
Realizację pokazano na poniższym zrzucie ekranu:



Powyższy skrypt zamieszczono na końcu dokumentu w wersji tekstowej (punkt 4.)

> **Uwaga!** Operacja na zmiennych użytych w trybie graficznym panelu nie powoduje odświeżania, dlatego w powyższym skrypcie zastosowano akcję ponownego wygenerowania strony!

- Na koniec należy dodać kolejne skrypty do wszystkich przycisków oraz używanych zdarzeń - odpowiednio: skrypt *ClickButton1* do zdarzenia *PANEL_BUTTON1->* `OnClick`.



3. Skrypt *Wyswietlacz* w wersji tekstowej:

```
    if(not (CLU_220000541->strona==1)) then
    if(not (CLU_220000541->strona==2)) then
    if(not (CLU_220000541->strona==3)) then
    if(CLU_220000541->strona>3) then
    CLU_220000541->strona=1
    CLU_220000541->Wyswietlacz()
    else
    if(CLU_220000541->strona<1) then
    CLU_220000541->strona=3
    CLU_220000541->Wyswietlacz()
    end
    end
    else
    CLU_220000541->timer_temp->Stop()
    CLU_220000541->x250000091_PANEL1->ClearScreen()
```

```
    CLU_220000541->x250000091_PANEL1->PrintText(63,31,"Witaj!",2)
    CLU_220000541->x250000091_PANEL1->DrawIcon(0,0,"onoff")
    CLU_220000541->x250000091_PANEL1->DisplayContent()
    end
    else
    CLU_220000541->x250000091_PANEL1->ClearScreen()
    CLU_220000541->timer_temp->Start()
    CLU_220000541->x250000091_PANEL1->PrintText(15,10,"Kuchnia [°C]:",2)
    CLU_220000541->x250000091_PANEL1->PrintFloat(80,38,CLU_220000541-
>x240000659_PANELSENSTEMP1->Value,1,2)
    CLU_220000541->x250000091_PANEL1->DrawLine(0,32,127,32,1)
    CLU_220000541->x250000091_PANEL1->DrawPoint(0,0,1)
    CLU_220000541->x250000091_PANEL1->DrawLine(70,32,70,63,1)
    CLU_220000541->x250000091_PANEL1->PrintText(15,40,CLU_220000541->Time,1)
    CLU_220000541->x250000091_PANEL1->DisplayContent()
    end
    else
    CLU_220000541->timer_temp->Stop()
    CLU_220000541->x250000091_PANEL_BUTTON1->SetIconA("lamp1off")
    CLU_220000541->x250000091_PANEL_BUTTON2->SetIconA("lamp2off")
    CLU_220000541->x250000091_PANEL_BUTTON3->SetIconA("shclosed")
    CLU_220000541->x250000091_PANEL_BUTTON4->SetIconA("winclose")
    CLU_220000541->x250000091_PANEL1->ShowButtons()
    end
```

4. Skrypt *ClickButton1* w wersji tekstowej:

```
    if(not (CLU_220000541->strona==1)) then
    if(not (CLU_220000541->strona==2)) then
    if(CLU_220000541->strona==3) then
    SYSTEM.Wait(1000)
    CLU_220000541->x250000091_PANEL_BUTTON1->ShowError()
    end
    else
    SYSTEM.Wait(1000)
    CLU_220000541->x250000091_PANEL_BUTTON1->ShowOK()
    end
    else
    if(CLU_220000541->x250000091_PANEL_BUTTON1->IconA=="lamp1off") then
    CLU_220000541->x210000904_DIMM1->SwitchOn(0,500)
    CLU_220000541->x250000091_PANEL_BUTTON1->SetIconA("lamp1on")
    else
    CLU_220000541->x210000904_DIMM1->SwitchOff(0,500)
    CLU_220000541->x250000091_PANEL_BUTTON1->SetIconA("lamp1off")
    end
    end
```

5. Konfiguracja modułu Smart Panel w wersji v4

> **Uwaga!** Smart Panel w wersji v4 dostępny jest dla Object Managera w wersji 1.2.1.190201 i wyższej oraz dala CLU z firmware 04.07.49-1912 i wyższym.

5.1. Parametry konfiguracyjne

A. Panel

CECHY

Nazwa	Opis
`GestureIconUp`	Nazwa pliku BMP z ikoną dla gestu Góra (bez rozszerzenia)
`GestureIconDown`	Nazwa pliku BMP z ikoną dla gestu Dół (bez rozszerzenia)
`GestureIconLeft`	Nazwa pliku BMP z ikoną dla gestu Lewo (bez rozszerzenia)
`GestureIconRight`	Nazwa pliku BMP z ikoną dla gestu Prawo (bez rozszerzenia)
`ProximitySens`	Czułość sensora zbliżeniowego (mniejsza wartość - większa czułość)
`ProximityTimeout`	Czas, po którym wyświetlacz zostanie wygaszony
`ProximityValue`	Sygnał sensora zbliżeniowego (wartość bezwymiarowa)
`BuzzerValue`	Sterowanie sygnalizacją dźwiękową: `0 - Off`, `1 - On`
`GestureMode`	Wybór orientacja gestów: `0 - Off`, `1 - Vertical`, `2 - Horizontal`, `3 - Vert+Horiz`
`GestureSens`	Wybór czułości gestów: `1 - Low`, `2 - Mid`, `3 - High`
`PageNr`	Numer wyświetlanej aktualnie strony
`PageDisplayMode`	Informacja przed zmianą strony: `0 - ShowImmediately`, `1 - ShowIconOrName`, `2 - ShowGesture`
`ButtonsLEDMode`	Lokalizacja przycisków za pomocą słabego światła LED: `0 - LocationLedOFF`, `1 - LocationLedON`, `2 - LocationLedONforActive`
`PageControlMode`	Źródło, które przełącza strony: `0 - Command` (przełączanie za pomocą metod SetNextPage oraz SetPrevPage) `1 - Gesture/Command` (przełączanie za pomocą gestów oraz metod SetNextPage i SetPrevPage)
`GestureDisplayMode`	Wyświetlanie informacji o aktualnie wykonanym geście: `0 - Off`, `1 - On`

METODY

Nazwa	Opis
`SwitchOnDisplay`	Wybudza wyświetlacz z trybu uśpienia
`ShowButtons`	Zmienia tryb wyświetlacza na *buttons*. Czyści wyświetlacz i wyświetla ponownie ikony (lub tekst) dla wszystkich przycisków
`ClearScreen`	Czyści zawartość wyświetlacza w trybie *freedraw*
`PrintText`	Wyświetla tekst w trybie *freedraw* z użyciem parametrów: `x`, `y`, `txt`, `font size`, gdzie: `x` oraz `y` to współrzędne wyrażone w pikselach, `txt` to string, `font size` to rozmiar czcionki (1:10p, 2:14p, 3:32p)
`PrintFloat`	Wyświetla liczbę w trybie *freedraw* z użyciem parametrów: `x`, `y`, `number`, `precision`, `font size`, gdzie: `x` oraz `y` to współrzędne wyrażone w pikselach, `number` to liczba, `precision` to ilość miejsc po przecinku, `font size` to rozmiar czcionki (1:10p, 2:14p, 3:32p)
`DrawLine`	Rysuje linie w trybie *freedraw* z użyciem parametrów: `x`, `y`, `xe`, `ye`, `color`, gdzie: `x` oraz to `y` współrzędne początkowe, `xe` oraz `ye` to współrzędne końcowe, `color` to kolor linii (0 - czarny, 1 - biały). Współrzędne początkowe oraz końcowe są wyrażone w pikselach
`DrawPoint`	Rysuje punkt w trybie *freedraw* z użyciem parametrów: `x`, `y`, `color`, gdzie: `x` oraz `y` to współrzędne wyrażone w pikselach, `color` to kolor punktu (0 - czarny, 1 - biały)
`DrawIcon`	Rysuje ikonę (bmp) w trybie *freedraw* z użyciem parametrów: `x`, `y`, `Filename`, gdzie: `x` oraz `y` to współrzędne wyrażone w pikselach, `Filename` to nazwa ikony (bez rozszerzenia)
`DisplayContent`	Wyświetla zawartość bufora pamięci graficznej. Zmienia tryb wyświetlacza na *freedraw*
`SetGestureIconUp`	Ustawia ikonę dla wykonania gestu w górę
`SetGestureIconDown`	Ustawia ikonę dla wykonania gestu w dół
`SetGestureIconLeft`	Ustawia ikonę dla wykonania gestu w lewo
`SetGestureIconRight`	Ustawia ikonę dla wykonania gestu w prawo
`SetProximitySens`	Ustawia wartość ProximitySens
`SetProximityTimeout`	Ustawia czas w sekundach po jakim wyświetlacz gaśnie
`SetBuzzerValue`	Sterowanie sygnalizacją dźwiękową (On/Off)

Nazwa	Opis
`SetGestureMode`	Wybór orientacji gestów
`SetGestureSens`	Wybór czułości gestów
`SetBeep`	Generuje dźwięk o danej częstotliwości[Hz], czasie trwania[ms] i głośności
`SetPageNr`	Ustawia numer wyświetlanej strony
`SetPageDisplayMode`	Ustawia tryb wyświetlania informacji przed zmianą strony
`SetButtonsLEDMode`	Ustawia tryb lokalizacji przycisków za pomocą diod LED
`SetPageControlMode`	Ustawia źródło, które przełącza strony (komendy/strony)
`SetGestureDisplayMode`	Ustawia tryb wyświetlania informacji o wykonanym geście
`SetNextPage`	Wyświetla następną stronę
`SetPrevPage`	Wyświetla poprzednią stronę
`Draw`	Wyzwala wywołanie zdarzenia OnDraw w przypadku, gdy OLED jest aktywny

ZDARZENIA

Nazwa	Opis
`OnGestureUp`	Zdarzenie wywoływane w przypadku wykonania gestu w górę
`OnGestureDown`	Zdarzenie wywoływane w przypadku wykonania gestu w dół
`OnGestureLeft`	Zdarzenie wywoływane w przypadku wykonania gestu w lewo
`OnGestureRight`	Zdarzenie wywoływane w przypadku wykonania gestu w prawo
`OnProximityDetect`	Zdarzenie wywołane w przypadku wykrycia zbliżającej się do wyświetlacza osoby
`OnPageChange`	Zdarzenie wywoływane w przypadku zmiany strony w panelu

B. Przyciski

CECHY

Nazwa	Opis
`Mode`	Zwraca ustawiony tryb działania przycisku: `0 - monostabilny (monostable)`, `1 - bistabilny (bistable)`, `2 - zablokowany (locked)`
`HoldDelay`	Czas w milisekundach, po jakim po wciśnięciu i przytrzymaniu przycisku wyzwalane jest zdarzenie `OnHold`
`HoldInterval`	Odstęp cykliczny w milisekundach, po jakim podczas trzymania przycisku wyzwalane jest zdarzenie `OnHold`
`Value`	Zwraca stan przycisku jako 0 lub 1
`Label`	Tekst, który opisuje przycisk (wyświetlany zamiast ikony)
`IconA`	Nazwa pliku ikony przypisanej do przycisku w trybie monostabilnym oraz bistabilnym w pozycji *OFF*; nazwa poprzedzona „~" wyświetli grafikę w negatywie; `IconA` ma priorytet nad cechą `Label`
`IconB`	Nazwa pliku ikony przypisanej do przycisku w trybie bistabilnym w pozycji *ON*; nazwa poprzedzona „~" wyświetli grafikę w negatywie

METODY

Nazwa	Opis
`SetMode`	Ustawia tryb działania przycisku: `0 - monostabilny (monostable)`, `1 - bistabilny (bistable)`, `2 - zablokowany (locked)`
`SetHoldDelay`	Ustawia wartość `HoldDelay`
`SetHoldInterval`	Ustawia wartość `HoldInterval`
`SetLabel`	Ustawia wartość `Label` (tekst opisujący przycisk)
`SetIconA`	Ustawia nazwę pliku ikony A (bez rozszerzenia)
`SetIconB`	Ustawia nazwę pliku ikony B (bez rozszerzenia)
`ShowOK`	Powoduje miganie zielonej diody na przycisku przez dwie sekundy (częstotliwość 500 ms). Czerwona dioda przycisku pozostaje zgaszona
`ShowError`	Powoduje miganie czerwonej diody na przycisku przez dwie sekundy (częstotliwość 500 ms). Zielona dioda przycisku pozostaje zgaszona
`LedSwitchOn`	Załącza zieloną diodę na przycisku
`RedLedSwitchOn`	Załącza czerwoną diodę na przycisku
`LedSwitchOff`	Wyłącza wszystkie diody na przycisku

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane przy zmianie stanu na przeciwny
`OnSwitchOn`	Zdarzenie wywoływane w momencie ustawienia stanu wysokiego na wejściu
`OnSwitchOff`	Zdarzenie wywoływane w momencie ustawienia stanu niskiego na wejściu
`OnShortPress`	Zdarzenie wywoływane po naciśnięciu przycisku na czas 500 ms - 2000 ms
`OnLongPress`	Zdarzenie wywoływane po naciśnięciu przycisku na czas 2000 ms – 5000 ms
`OnHold`	Zdarzenie wywoływane, gdy wejście jest w stanie wysokim, pierwszy raz po upłynięciu czasu `HoldDelay`, a następnie cyklicznie co wartość `HoldInterval`
`OnClick`	Zdarzenie wywoływane po naciśnięciu przycisku na czas krótszy niż 500ms

C. Konfiguracja stron (Panel_Page)

CECHY

Nazwa	Opis
`PageType`	Typ strony wyświetlanej na Smart Panelu: `0 - Inactive`, `1 - Buttons`, `2 - Thermostats`, `3 - FreeDraw`
`PageName`	Nazwa strony/nazwa ikony wyświetlana na Smart Panelu (przy przechodzeniu pomiędzy stronami)
`Object_1_Id`	ID obiektu termostatu lub numer przycisku w zależności od typu strony, np.: Dla strony typu *Thermostats*: - w przypadku termostatu na lokalnym CLU: THE1325 - w przypadku termostatu na zdalnym CLU: CLU220000001->THE4321 W przypadku cechy PageType ustawionej na *Buttons*/*FreeDraw* należy wpisać numer przycisku (1..16)
`Object_1_Name`	Nazwa termostatu wyświetlana na stronie Smart Panelu. Dotyczy tylko strony *Thermostats* (brak nazwy - termostat nieaktywny). W przypadku cechy `PageType` ustawionej na *Buttons*/*FreeDraw* cecha pozostaje pusta
`Object_2_Id`	ID obiektu termostatu lub numer przycisku w zależności od typu strony, np.: Dla strony typu *Thermostats*: - w przypadku termostatu na lokalnym CLU: THE1325 - w przypadku termostatu na zdalnym CLU: CLU220000001->THE4321 W przypadku cechy PageType ustawionej na *Buttons*/*FreeDraw* należy wpisać numer przycisku (1..16)
`Object_2_Name`	Nazwa termostatu wyświetlana na stronie Smart Panelu. Dotyczy tylko strony *Thermostats* (brak nazwy - termostat nieaktywny). W przypadku cechy `PageType` ustawionej na *Buttons*/*FreeDraw* cecha pozostaje pusta
`Object_3_Id`	ID obiektu termostatu lub numer przycisku w zależności od typu strony, np.: Dla strony typu *Thermostats*: - w przypadku termostatu na lokalnym CLU: THE1325 - w przypadku termostatu na zdalnym CLU: CLU220000001->THE4321 W przypadku cechy PageType ustawionej na *Buttons*/*FreeDraw* należy wpisać numer przycisku (1..16)
`Object_3_Name`	Nazwa termostatu wyświetlana na stronie Smart Panelu. Dotyczy tylko strony *Thermostats* (brak nazwy - termostat nieaktywny). W przypadku cechy `PageType` ustawionej na *Buttons*/*FreeDraw* cecha pozostaje pusta
`Object_4_Id`	ID obiektu termostatu lub numer przycisku w zależności od typu strony, np.: Dla strony typu *Thermostats*: - w przypadku termostatu na lokalnym CLU: THE1325 - w przypadku termostatu na zdalnym CLU: CLU220000001->THE4321 W przypadku cechy PageType ustawionej na *Buttons*/*FreeDraw* należy wpisać numer przycisku (1..16)
`Object_4_Name`	Nazwa termostatu wyświetlana na stronie Smart Panelu. Dotyczy tylko strony *Thermostats* (brak nazwy - termostat nieaktywny). W przypadku cechy `PageType` ustawionej na *Buttons*/*FreeDraw* cecha pozostaje pusta

METODY

Nazwa	Opis
`SetPageType`	Ustawia typ strony wyświetlanej na Smart Panelu
`SetPageName`	Ustawia nazwę strony/nazwę ikony wyświetlanej na Smart Panelu (przy przechodzeniu pomiędzy stronami)
`SetObject_1_Id`	Ustawia wartość `Object_1_Id`
`SetObject_1_Name`	Ustawia wartość `Object_1_Name`
`SetObject_2_Id`	Ustawia wartość `Object_2_Id`
`SetObject_2_Name`	Ustawia wartość `Object_2_Name`
`SetObject_3_Id`	Ustawia wartość `Object_3_Id`
`SetObject_3_Name`	Ustawia wartość `Object_3_Name`
`SetObject_4_Id`	Ustawia wartość `Object_4_Id`
`SetObject_4_Name`	Ustawia wartość `Object_4_Name`

ZDARZENIA

Nazwa	Opis
`OnPageOpen`	Zdarzenie wywoływane po otwarciu strony
`OnPageClose`	Zdarzenie wywoływane po zamknięciu strony
`OnDraw`	Zdarzenie sygnalizujące potrzebę przerysowania. Generowanie wyłącznie w trybie pracy *freedraw*, po wejściu na daną stronę lub w momencie wywołania metody `Draw` i wybudzeniu ekranu

D. Czujniki temperatury i oświetlenia

CECHY

Nazwa	Opis
`Threshold`	Wielkość histerezy (dokładność 0.1°C / 0.1 %) określająca czułość, przy której następuje wygenerowanie zdarzeń: `OnChange`, `OnLowerValue`, `OnRaiseValue`
`Sensitivity`	Okres (w ms), w którym próbkowane wartości są uśredniane
`MinValue`	Minimalna wartość cechy `Value`, której przekroczenie wywołuje zdarzenie `OnOutOfRange`
`MaxValue`	Maksymalna wartość cechy `Value`, której przekroczenie wywołuje zdarzenie `OnOutOfRange`
`Value`	Wartość wejścia: dla czujnika temperatury od 0.0 do 45.0°C lub dla czujnika światła 0 – 100%

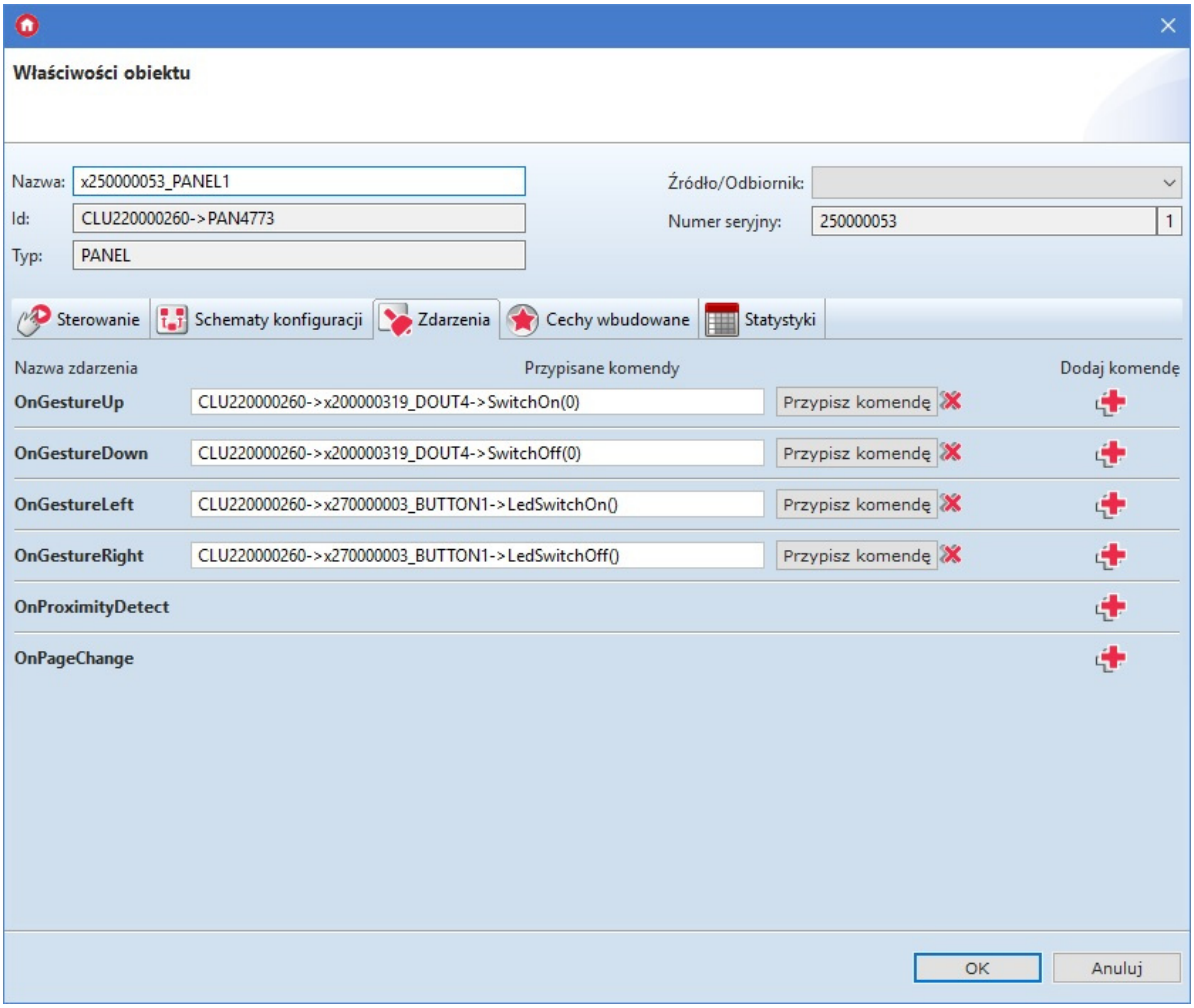
ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w przypadku zmiany wartości cechy Value
`OnRaiseValue`	Zdarzenie wywoływane przy zmianie wartości na wyższą (zbocze narastające)
`OnLowerValue`	Zdarzenie wywoływane przy zmianie wartości na niższą (zbocze opadające)
`OnOutOfRange`	Zdarzenie wywoływane, gdy wartość na wejściu znajduje się poza wyznaczonym zakresem (`MinValue`; `MaxValue`)

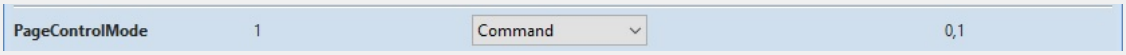
5.2. Tworzenie konfiguracji sensora gestów

Chcąc utworzyć konfigurację dla sensora gestów należy:

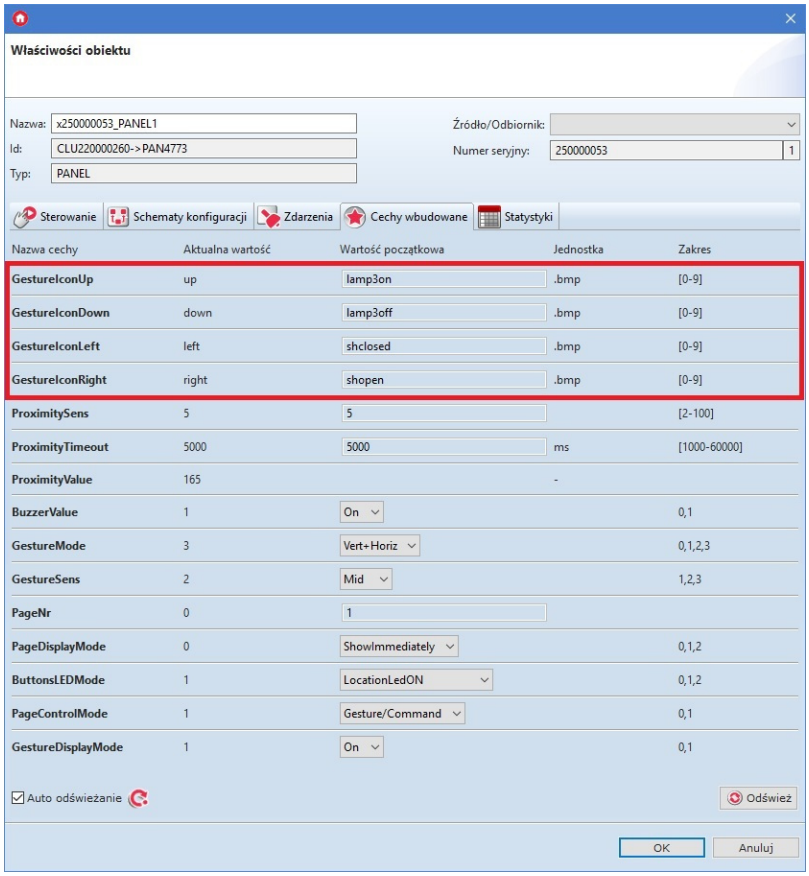
- Otworzyć - przez dwuklik - obiekt *Panel*;
- Przejść do zakładki *Zdarzenia*;
    - Przypisać metody do zdarzeń `OnGestureUp`, `OnGestureDown`, `OnGestureLeft`, `OnGestureRight` (klikając w `+` po prawej stronie każdej z metod):



Uwaga! W przypadku konfiguracji zawierającej konfigurację stron (Buttons/FreeDraw/Thermostats) metody przypisane do zdarzeń OnGestureLeft oraz OnGestureRight nie będą wykonywane. Jest to związane z predefiniowaną funkcjonalnością polegającą na przechodzeniu pomiędzy stronami. Istnieje możliwość zmiany sposobu przewijania stron. W tym celu należy zmienić ustawienie cechy `PageControlMode` na wartość Command. Po wykonaniu tej czynności metody przypisane do zdarzeń będą wykonywane.

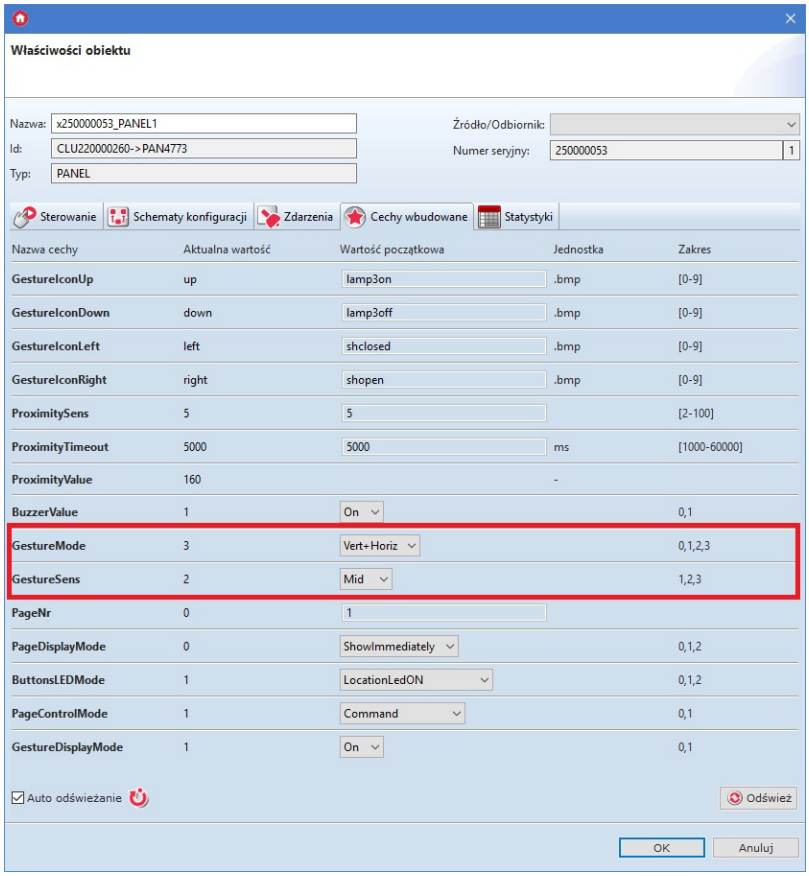


Możliwa jest również podmiana domyślnie wyświetlanych ikon przy wykonywaniu gestów - w tym celu należy przejść do zakładki *Cechy wbudowane* i wpisać nazwy żądanych ikon bez rozszerzenia *.bmp:*



Użycie ikon będzie możliwe, gdy będą wgrane na kartę microSD z rozszerzeniem *.bmp*.

Dodatkowo od wersji 04.03.04.1910 istnieje możliwość wyboru orientacji rozpoznawanych gestów oraz ich czułości. W tym celu należy przejść do zakładki *Cechy wbudowane* i wybrać żądaną orientację i czułość rozpoznawania gestów:



Cechy wbudowane za pośrednictwem których można dokonywać wyboru orientacji i czułości to:

- `GestureMode` - możliwa zmiana kierunku detekcji gestu:
  - Off - gesty nie są rozpoznawane;
  - Vertical - rozpoznawane są wyłącznie gesty w górę oraz w dół;
  - Horizontal - rozpoznawane są wyłącznie gesty w lewo oraz w prawo;
  - Vert+Horiz - rozpoznawane są gesty zarówno w górę, w dół, jak również w lewo oraz w prawo.
- `GestureSens` - możliwa zmiana czułości wykrywania gestów:
  - Low - gest wykonywany blisko urządzenia w sposób dokładny;
  - Mid - gest wykonywany zarówno blisko urządzenia, jak również z niewielkiej odległości;
  - High - gest wykonywany z dalszej odległości, istnieje możliwość detekcji błędnego gestu.

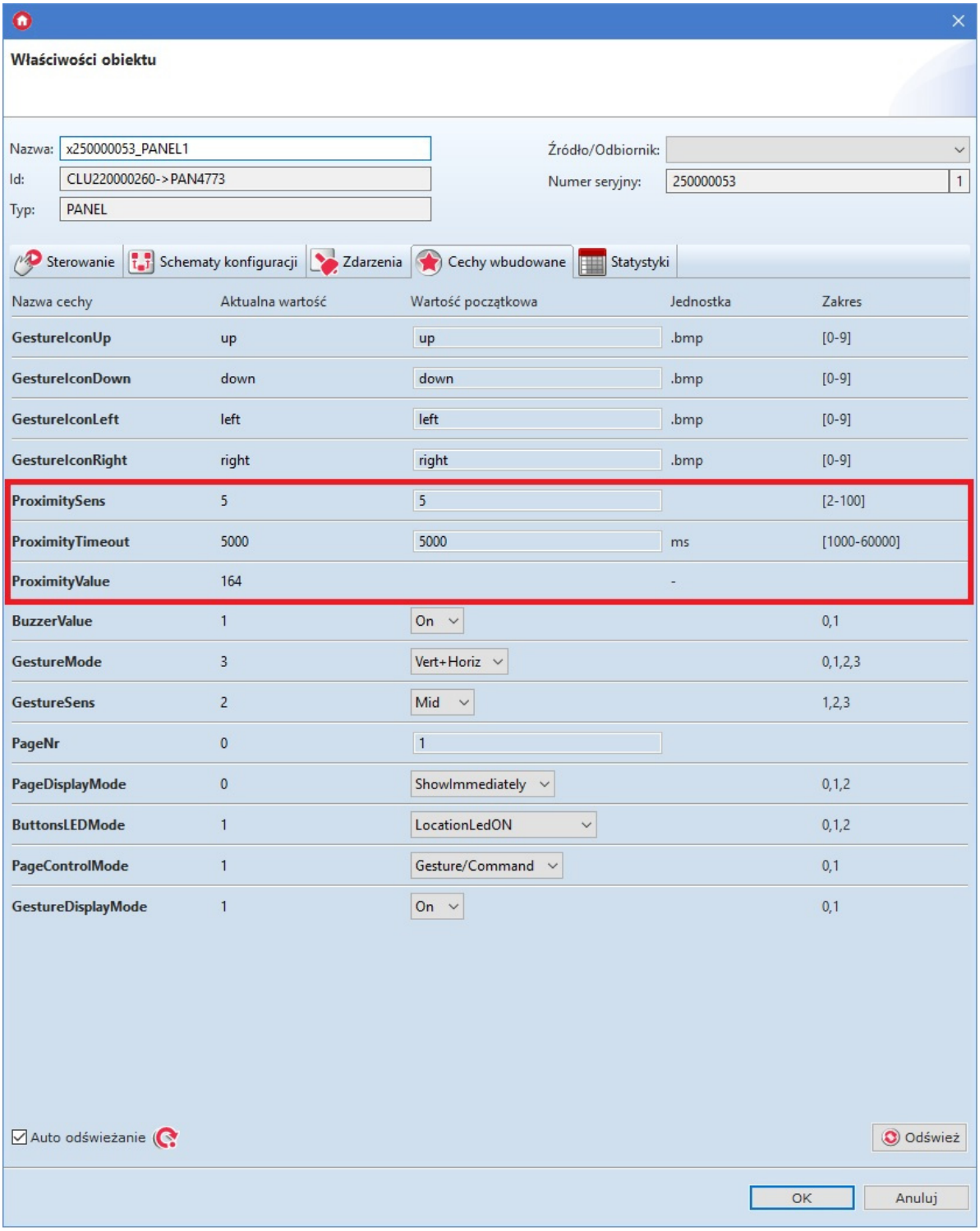
Powyższe cechy można ustawiać zarówno w zakładce *Cechy wbudowane*, jak również przy pomocy metod: `SetGestureIconUp`, `SetGestureIconDown`, `SetGestureIconLeft`, `SetGestureIconRight`, `SetGestureMode`, `SetGestureSens` (w metodach obiektu Panel).

- Zatwierdzić okno konfiguracji przyciskiem *OK*;
- Wysłać konfigurację do CLU Z-Wave.

5.3. Konfiguracja czujnika zbliżenia

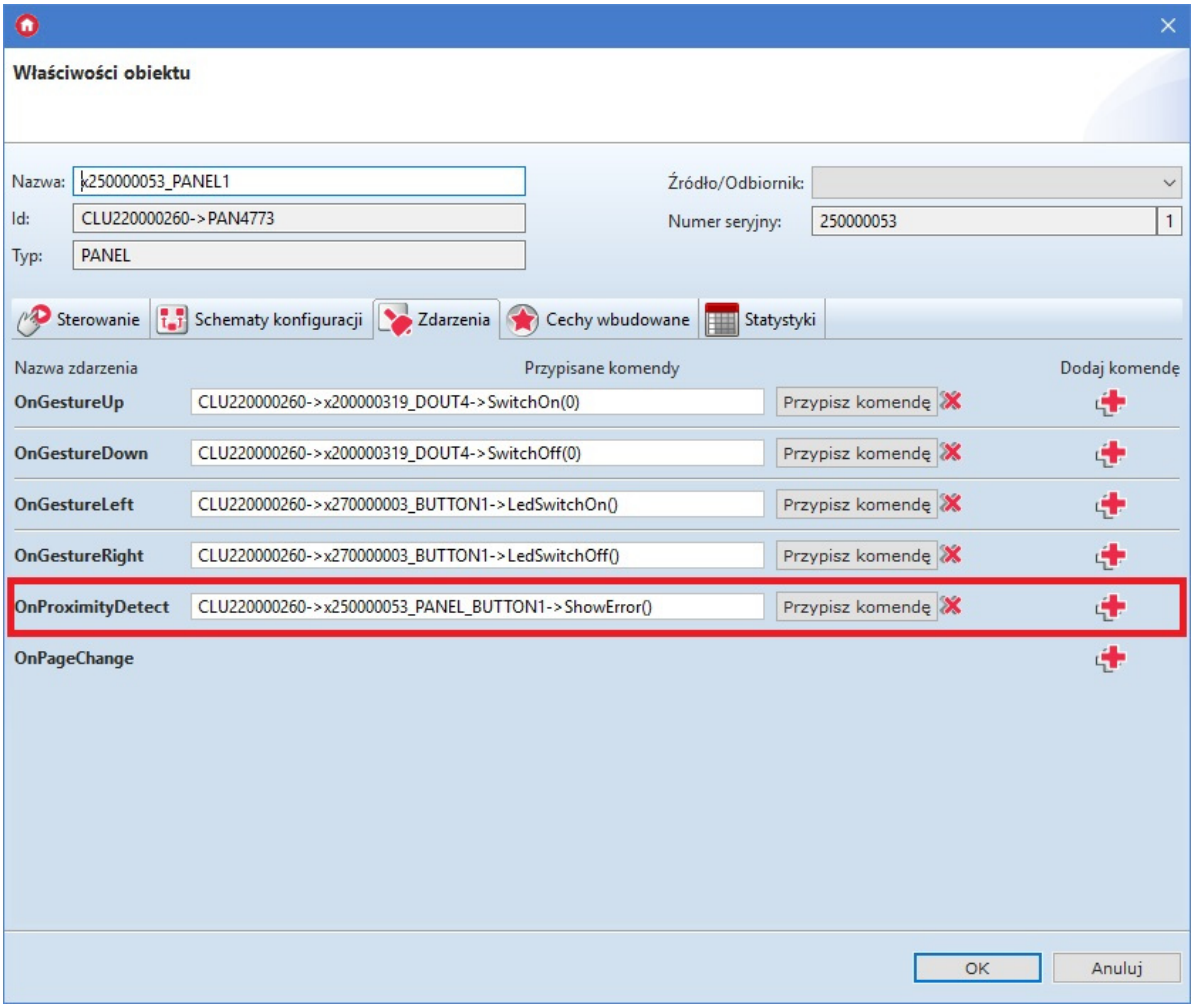
W celu ustawienia parametrów czujnika zbliżenia należy:

- Otworzyć - przez dwuklik - obiekt Panel;
- Przejść do zakładki *Cechy wbudowane*, gdzie znajdują się 3 cechy odnoszące się do czujnika zbliżenia:
  - `ProximitySens` - określa czułość sensora;
  - `ProximityTimeout` - definiuje czas, po jakim wyświetlacz zostaje wygaszony, gdy nie zostanie wykryty ruch;
  - `ProximityValue` - zwraca przybliżoną odległość w centymetrach od panelu do obiektu;



Powyższe cechy można ustawiać zarówno w zakładce *Cechy wbudowane*, jak również przy pomocy metod: `SetProximitySens` oraz `SetProximityTimeout` (w metodach obiektu Panel).

- Reakcja czujnika zbliżenia generuje zdarzenie `OnProximityDetect`, do którego można dodać dodatkowe metody:



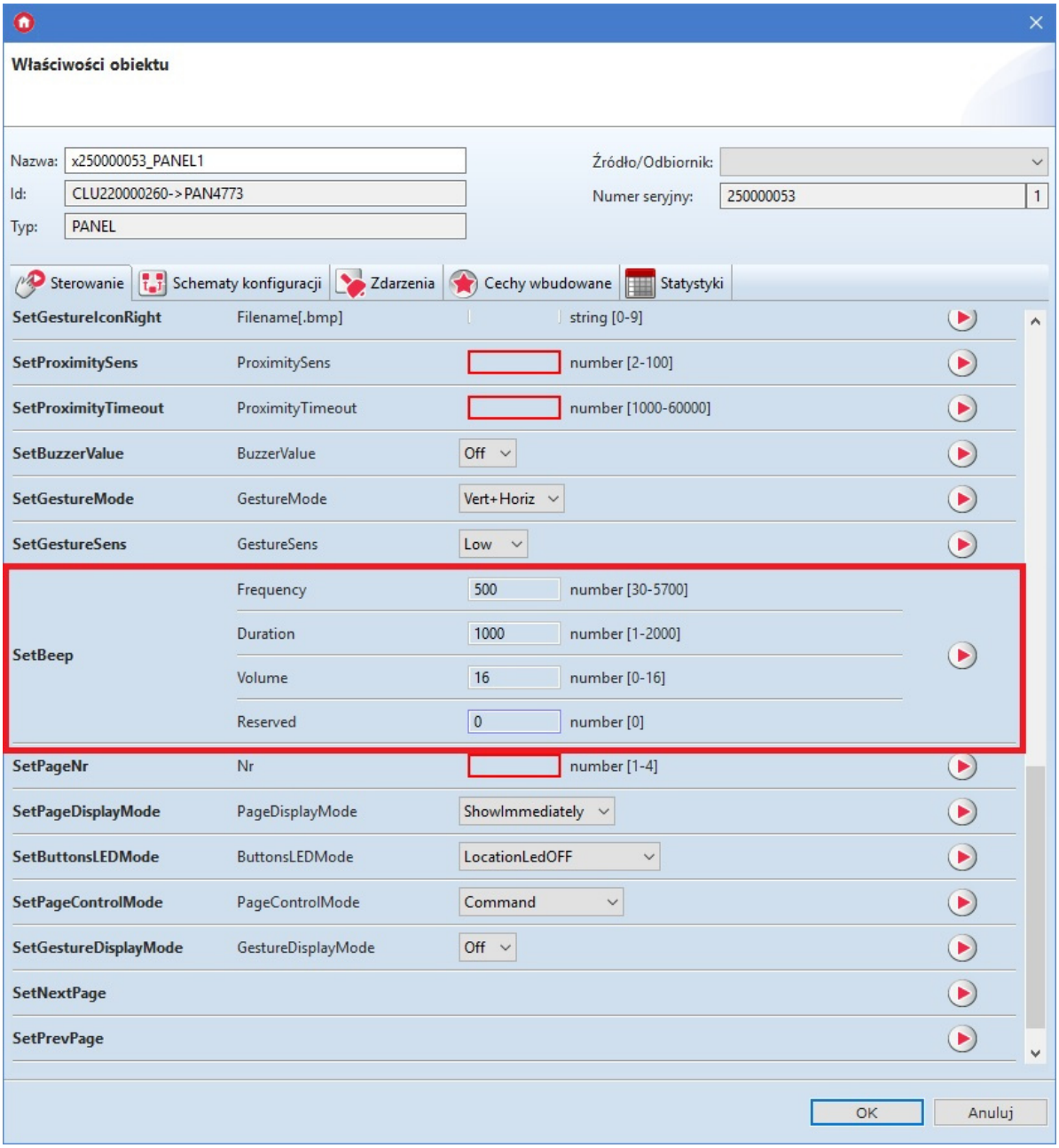
- Wysłać konfigurację do CLU Z-Wave.

5.4. Obiekt Panel - nowa funkcjonalność

W najnowszej wersji modułu Smart Panel (od 04.03.04.1910) w obiekcie Panel wprowadzona została nowa funkcjonalność umożliwiająca między innymi:

- wygenerowanie dźwięku;
- zarządzanie podświetleniem LED przycisków;
- możliwość włączenia/wyłączenia informowania o wykrytym geście;
- mechanizm zarządzania stronami, który zostanie opisany dokładnie w kolejnym podrozdziale.

Pierwszą z wprowadzonych nowości jest możliwość wygenerowania dźwięku o zadanej częstotliwości, długości oraz głośności. Do tego celu służy metoda `SetBeep`:



Kolejną funkcjonalnością dostępną od najnowszej wersji oprogramowania jest możliwość lokalizacji przycisków za pomocą słabego światła LED. W tym celu należy przejść do zakładki *Cechy wbudowane* i ustawić żądaną wartość cechy `ButtonsLEDMode`:

- LocationLedOFF - przyciski na module Smart Panel są nie podświetlone;
- LocationLedOn - przyciski na module SmartPanel są lekko podświetlone;
- LocationLedforActive - podświetlone zostają wyłącznie przyciski, które znajduje się w jednym z dwóch trybów pracy *Monostable/Bistable*. Jeżeli przycisk znajduje się w trybie *Locked*, jego dioda pozostaje zgaszona.

Oprócz możliwości zarządzania podświetleniem przycisków, istnieje możliwość włączenia/wyłączenie informowania o wykryciu gestu. W tym celu w zakładce *Cechy wbudowane* odnaleźć cechę `GestureDisplayMode`, ustawiając żądaną wartość:

- Off - informacja o wykryciu gestu nie jest wyświetlana na ekranie modułu;
- On - informacja o wykryciu gestu jest wyświetlana na ekranie modułu.

Powyższe cechy wbudowane można ustawiać również przy pomocy metod: `SetButtonsLEDMode` oraz `SetGestureDisplayMode`.

5.5. Obiekt Panel - mechanizm zarządzania stronami

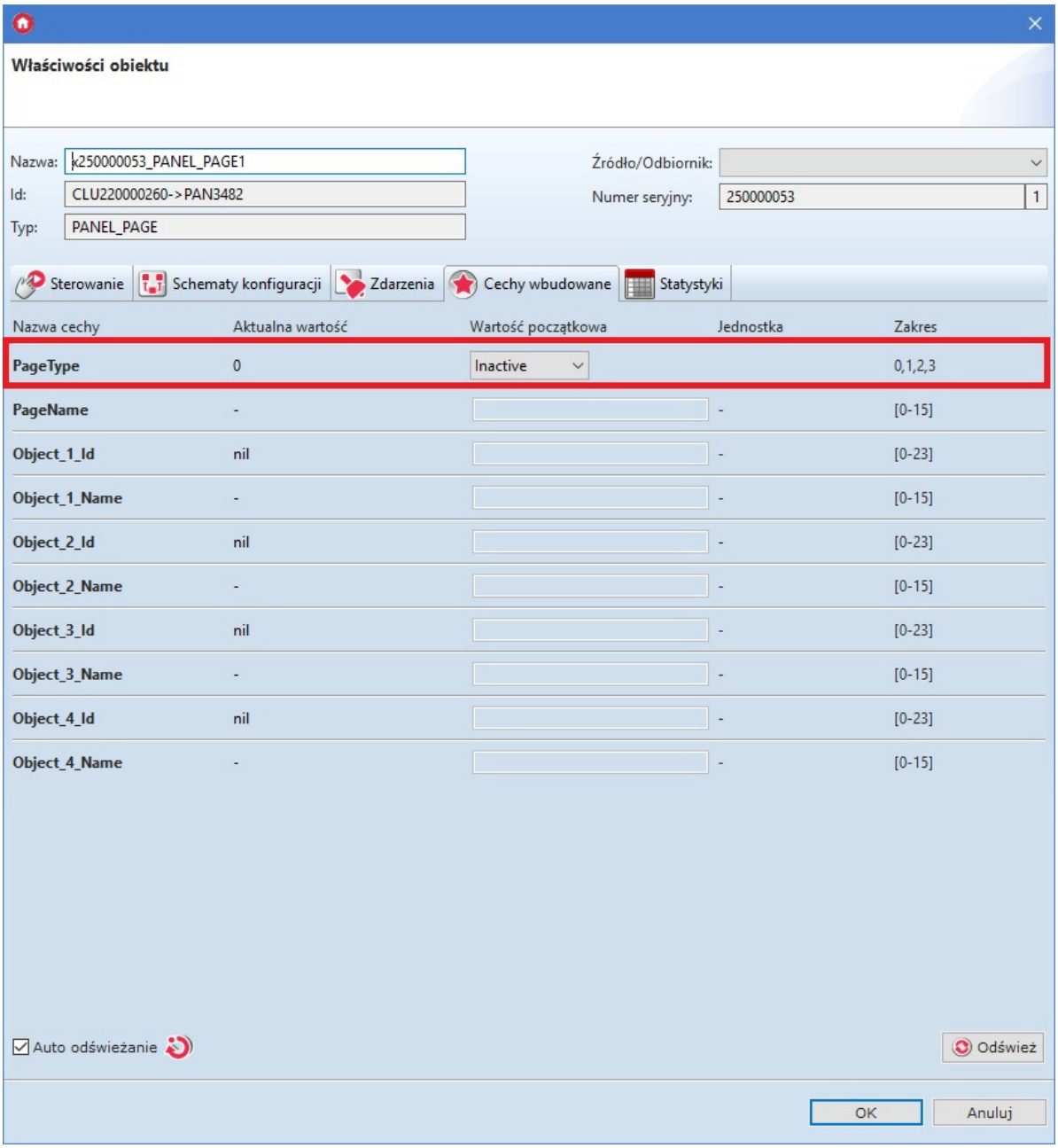
Smart Panel w wersji v4 wprowadza nowy mechanizm zarządzania stronami. W jego skład wchodzą cechy, metody oraz zdarzenie, które umieszczone zostały w obiekcie Panel:

- Metody/Cechy:
  - `SetPageNr/PageNr` - za pomocą niniejszej metody/cechy istnieje możliwość bezpośredniego przejścia pomiędzy większą ilością stron jednocześnie. Podając w parametrze numer strony, a następnie wywołując metodę, na ekranie wyświetlona zostanie żądana strona (możliwa potrzeba wybudzenia ekranu);
  - `SetPageDisplayMode/PageDisplayMode` - za pośrednictwem metody/cechy istnieje możliwość ustawienia sposobu przechodzenia pomiędzy stronami. Do wyboru jest trzy tryby:
    - ShowImmediately (0) - przejście pomiędzy stronami odbywa się natychmiastowo, nie jest poprzedzone wyświetleniem komunikatu/ikony/nazwy;
    - ShowIconOrName (1) - przejście pomiędzy stronami poprzedza wyświetlenie ikony lub nazwy wprowadzonej w cesze `PageName`;
    - ShowGesture (2) - przejście pomiędzy stronami poprzedza wyświetlenie ikony wprowadzonej w cesze `GestureIconLeft` lub `GestureIconRight`, w zależności od wykonanego gestu;
  - `SetPageControlMode/PageControlMode` - przy użyciu metody/cechy istnieje możliwość zmiany źródła, za pomocą którego dokonywana jest zmiana strony:
    - Command (0) - przejście do poprzedniej/następnej strony następuje wyłącznie za pomocą metod `SetPrevPage` oraz `SetNextPage`. Dodatkowo gesty w lewo oraz w prawo stają się aktywne, co oznacza, iż istnieje możliwość przypisania do zdarzeń `OnGestureLeft` oraz `OnGestureRight` akcji;
    - Gesture/Command (1) - przejście do poprzedniej/następnej strony możliwe jest za pomocą gestów w lewo oraz w prawo, jak również z wykorzystaniem metod `SetPrevPage` oraz `SetNextPage`. W przypadku ustawienia takiej wartości cechy, gesty w lewo oraz w prawo posiadają predefiniowaną funkcjonalność, która ma wyższy priorytet nad akcjami przypisanymi do zdarzeń `OnGestureLeft` oraz `OnGestureRight`. Oznacza to, iż akcje przypisane do tych zdarzeń nie będą wykonywane;
  - `SetNextPage` - metoda umożliwia przejście do następnej strony w konfiguracji;
  - `SetPrevPage` - metoda umożliwia przejście do poprzedniej strony w konfiguracji;
  - `Draw` - metoda służąca do wygenerowania zdarzenia OnDraw w momencie, kiedy OLED jest aktywny;
- Zdarzenie:
  - `OnPageChange` - zdarzenie generowane w momencie przejścia pomiędzy stronami

> **Uwaga!** Mechanizm zarządzania stronami dostępny jest wyłącznie dla konfiguracji stron wykonanych za pośrednictwem obiektów Panel_Page (Buttons/FreeDraw/Thermostats). W przypadku konfiguracji, która została stworzona w sposób dotychczasowy (podrozdział 4.5), powyższe cechy, metody oraz zdarzenie są ignorowane.

5.6. Kompatybilność wstecz

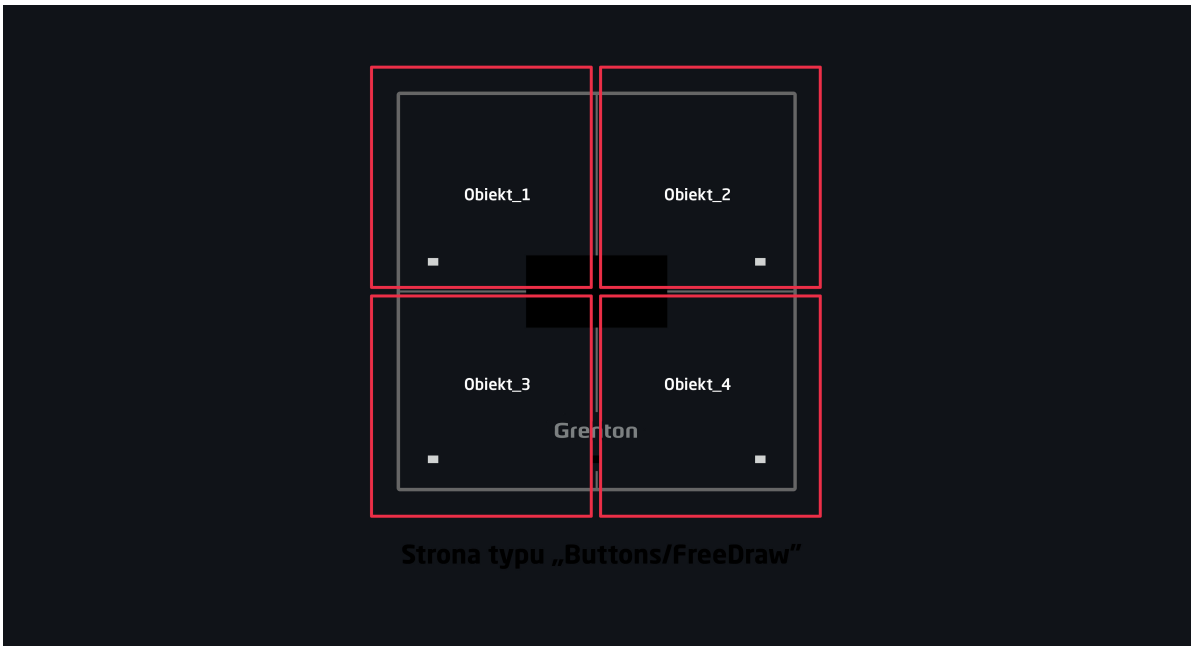
Rozpoczynając pracę z nową wersją modułu Smart Panel, urządzenie znajduje się domyślnej konfiguracji, która jest kompatybilna wstecz. Wszystkie cztery obiekty Panel_Page posiadają ustawioną cechę wbudowaną `PageType` na wartość *Inactive*. Umożliwia to pracę z panelem w taki sam sposób jak dotychczas (w wersji v3). Dostępne są wyłącznie pierwsze cztery przyciski widoczne na liście obiektów. Przyciski od 5 do 16 pomimo możliwości konfiguracji są nieaktywne. Tworzenie konfiguracji wielu stron odbywa się zgodnie z procedurą opisaną w podrozdziale 4.5.



5.7. Tworzenie konfiguracji z wykorzystaniem obiektu strony Buttons

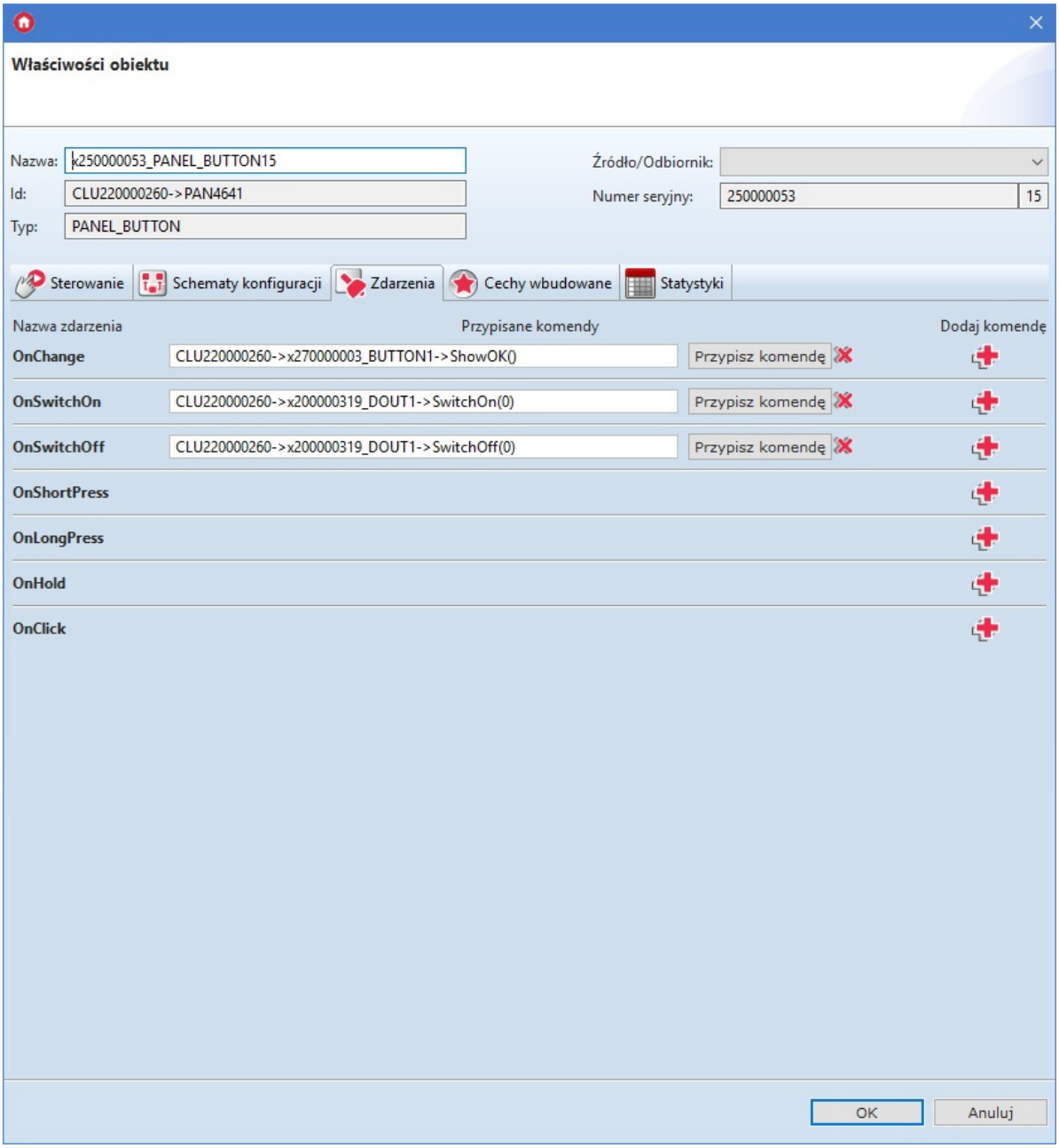
W trybie pracy *Buttons* do dyspozycji jest 4 fizyczne przyciski dotykowe i do 16 przycisków wirtualnych rozłożonych na 4 stronach, z których każdy może realizować niezależne funkcje. Istnieje również możliwość łączenia/scalania 2,3,4 obiektów w jeden przycisk (szerzej opisane w podrozdziale XII.5.10).

Uwaga! W trybie pracy *Buttons* rysowanie treści na wyświetlaczu jest zablokowane.

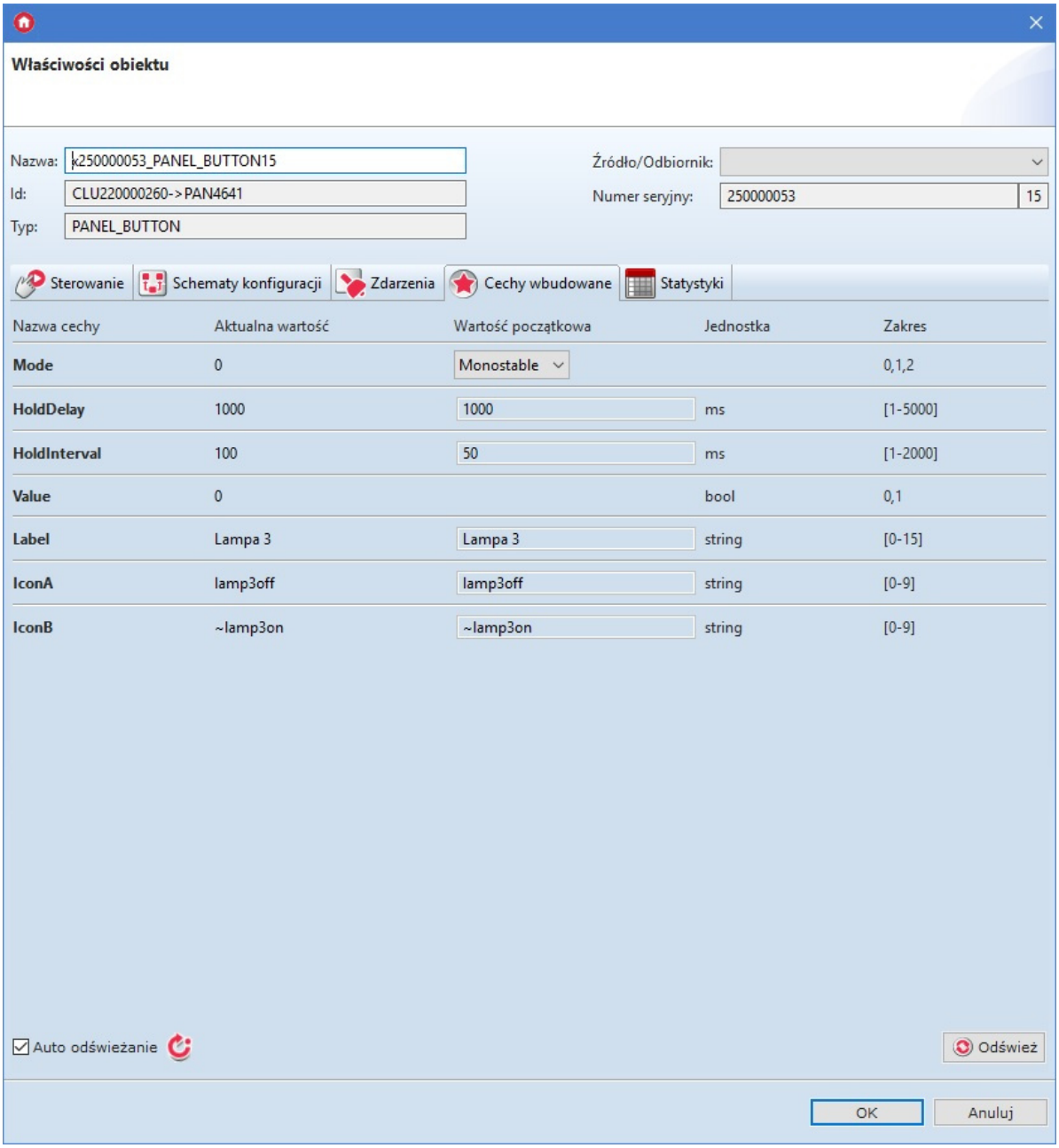


Tworzenie konfiguracji panelu obsługującej stronę lub strony typu *Buttons* najlepiej rozpocząć od konfiguracji przycisków, które mają zostać wykorzystane. W celu ich parametryzacji należy:

- Otworzyć obiekt *PANEL_BUTTONX* (gdzie X to numer jednego z 16 przycisków) poprzez dwukrotne kliknięcie na liście modułów;
- Przejście do zakładki Zdarzenia;
- Skonfigurować działanie przycisku poprzez przypisanie metod do określonych zdarzeń (klikając ”+” po prawej stronie okna):



- Wybrać zakładkę *Cechy wbudowane* i zdefiniować obiekty wyświetlane na ekranie danego przycisku:
    - `Label` - cecha określająca tekst przypisany do danego przycisku;
    - `IconA` - cecha określająca nazwę ikony przypisanej do danego przycisku, gdy znajduje się w trybie *Monostable*, bądź dla tryb *Bistable* dla pozycji OFF;
    - `IconB` - cecha określająca nazwę ikony przypisanej do danego przycisku, gdy znajduje się w trybie *Bistable* w pozycji ON. Chcąc przypisać tę samą ikonę, ale o odwróconej kolorystyce należy nazwę piktogramu poprzedzić znakiem "~" (np. `~heaton`):



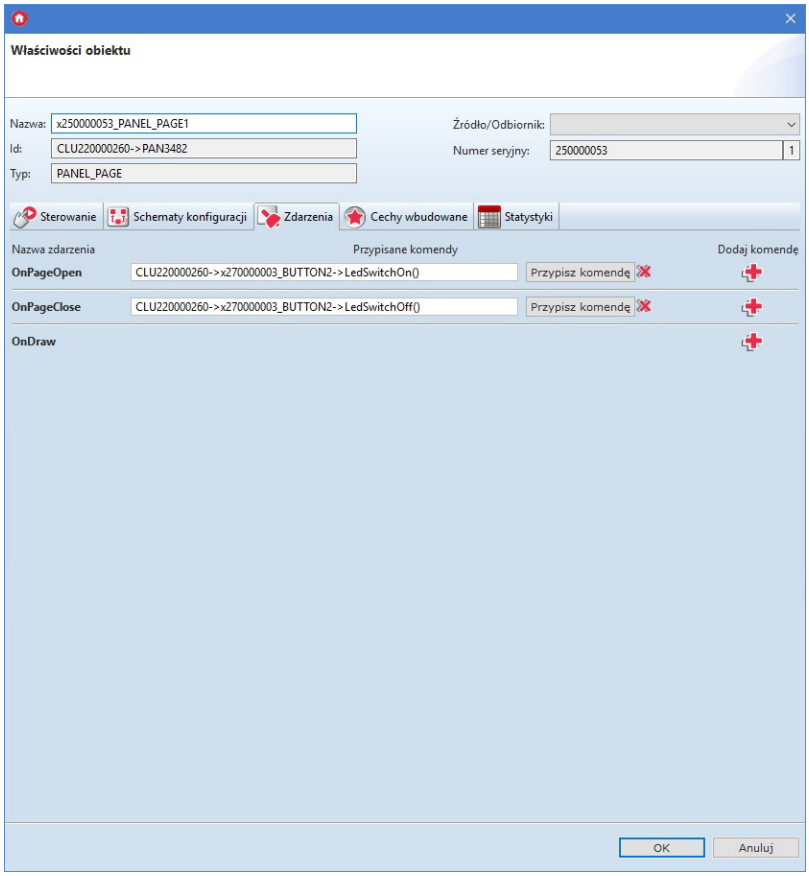
Powyższe cechy wbudowane można ustawić zarówno w zakładce *Cechy wbudowane*, jak również za pośrednictwem metod: `SetLabel`, `SetIconA`, `SetIconB`.

> **Uwaga!** Metoda `SetIconA` ma w systemie wyższy priorytet niż metoda `SetLabel`!

- Wysłać konfigurację do CLU Z-Wave.

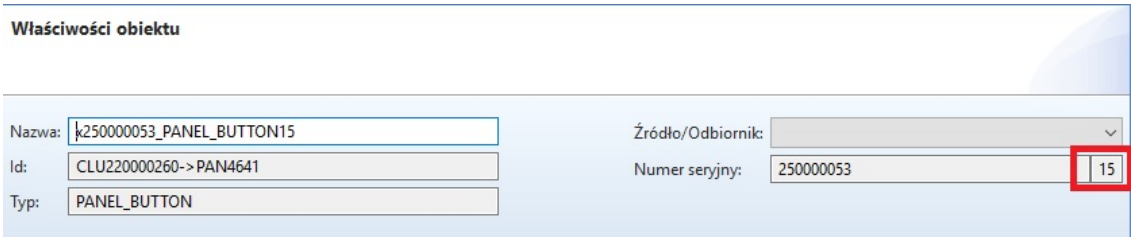
Kolejnym krokiem w tworzeniu konfiguracji jest konfiguracja obiektów Panel_Page w zależności od ilości przycisków. Jeden obiekt Panel_Page obsługuje do 4 przycisków. W tym celu należy:

- Otworzyć obiekt *PANEL_PAGEX* (gdzie X to numer kolejnej strony) poprzez dwukrotne kliknięcie na liście modułów;
- Przejście do zakładki *Zdarzenia*;
- Skonfigurować działanie strony poprzez przypisanie metod do określonych zdarzeń (klikając w "+" po prawej stronie okna):

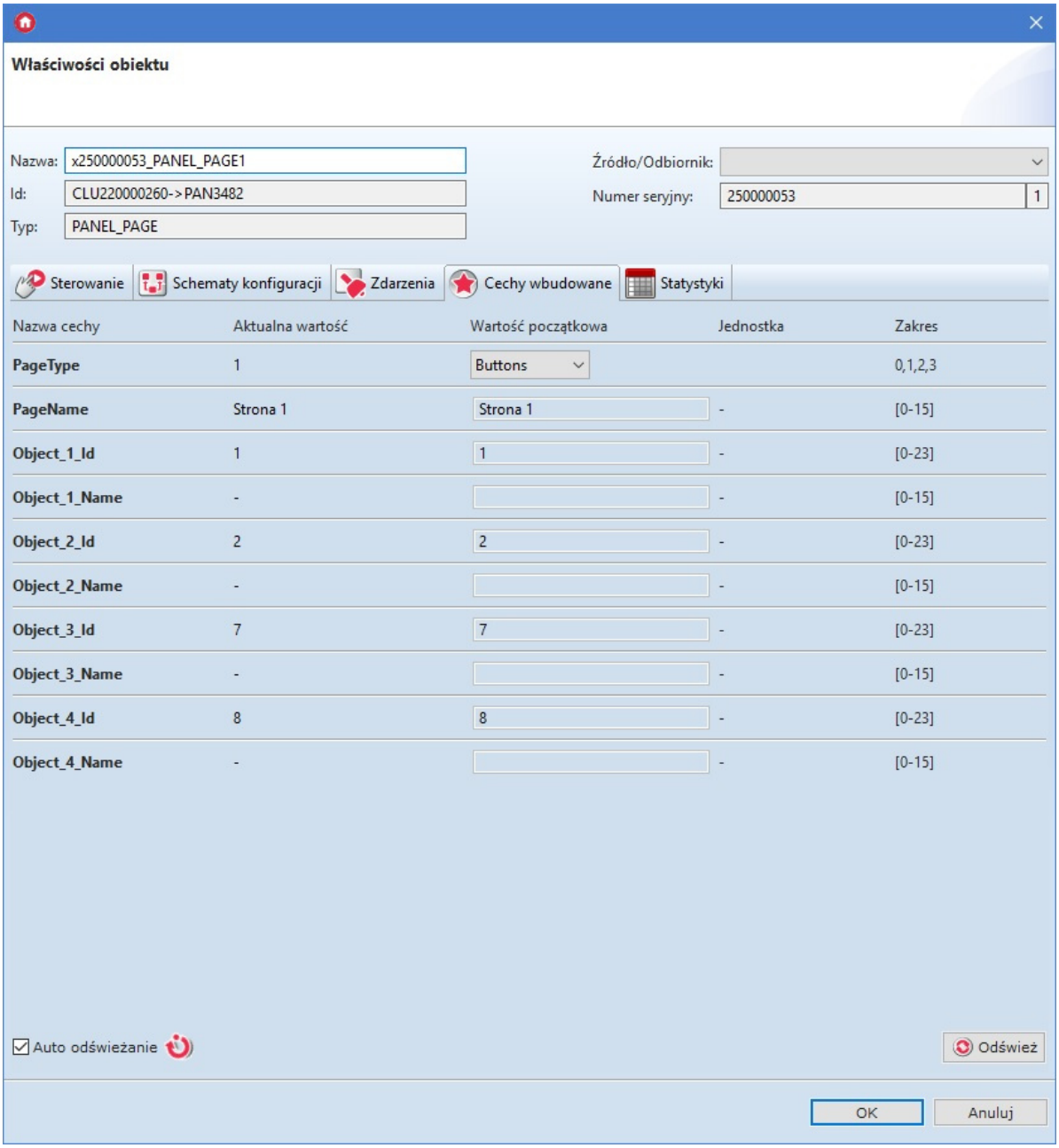


Uwaga! W przypadku typu strony *Buttons* zdarzenie `OnDraw` nie jest generowane.

- Wybrać zakładkę *Cechy wbudowane* i zdefiniować obsługiwany typ strony i powiązać obiekty strony z przyciskami:
  - `PageType` - cecha określająca typ strony, należy ustawić ją na wartość *Buttons (1)*;
  - `PageName` - cecha określająca nazwę strony lub ikonę, która będzie wyświetlana w momencie przechodzenia pomiędzy stronami (działa wyłącznie przy ustawionej cesze `PageDisplayMode` na wartość 1 (ShowIconOrName) w obiekcie Panel);
  - `Object_X_Id` - identyfikator/numer przycisku. W celu należy odczytać wartość znajdującą się w polu *Numer seryjny* obiektu *PANEL_BUTTONX*



  - `Object_X_Name` - nazwa termostatu. W przypadku typu strony *Buttons* cechę należy pozostawić pustą;

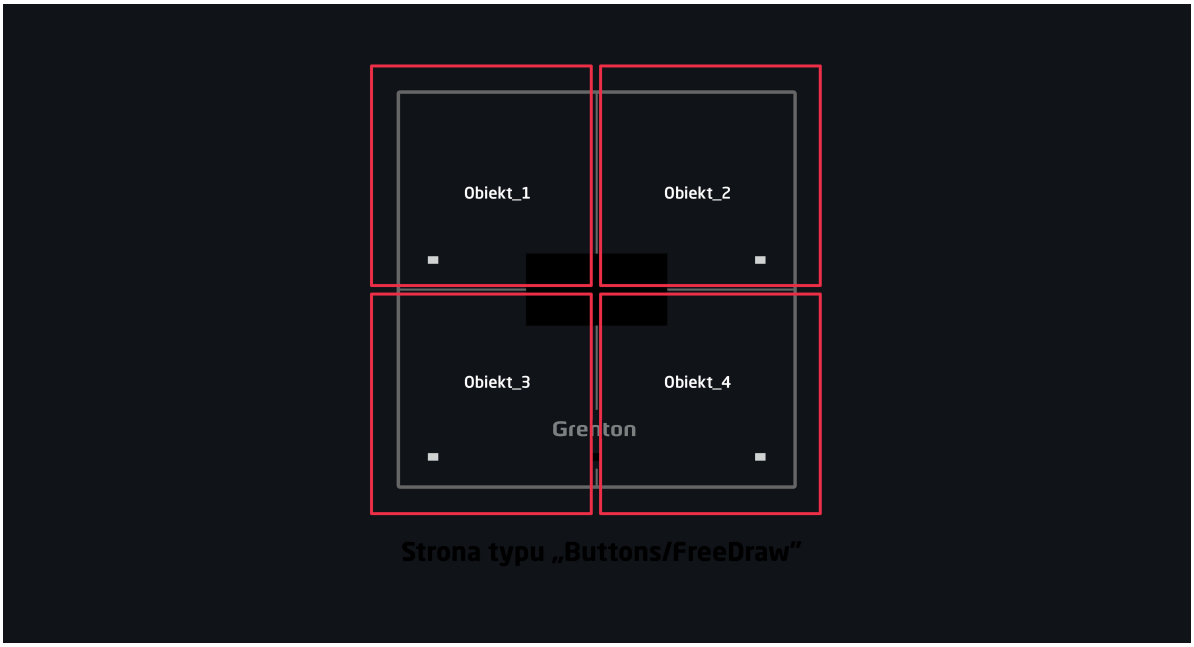


Uwaga! Wysłanie konfiguracji wyłącznie ze zdefiniowanym typ strony, bez ustawionych powiązań obiektów z przyciskami wiąże się z uruchomieniem trybu pracy panelu jako *Buttons*. Jednakże przyciski na module będą nieaktywne. Ma to związek z nie uzupełnieniem cech Object_X_Id.

- Wysłać konfigurację do CLU Z-Wave.

5.8. Tworzenie konfiguracji z wykorzystaniem obiektu strony FreeDraw

W trybie pracy *FreeDraw* podobnie jak w przypadku *Buttons* do dyspozycji są 4 fizyczne przyciski dotykowe i do 16 przycisków wirtualnych rozłożonych na 4 stronach, z których każdy może realizować niezależne funkcje. Istnieje również możliwość łączenia/scalania obiektów w jeden przycisk. Wyświetlacz OLED działa w trybie *FreeDraw*, czyli jest w pełni dostępny dla skryptów LUA użytkownika. Stworzony został również mechanizm rysowania, w którym skrypty rysujące wywoływane są zdarzeniem `OnDraw` generowanym przez panel w momencie, gdy jest to konieczne. System wywołuje metodę `Draw` w momencie, kiedy rysowany na module kontent uległ zmianie.



A. Ogólne zasady tworzenia konfiguracji

Tworzenie konfiguracji panelu obsługującej stronę lub strony typu *FreeDraw* najlepiej rozpocząć od konfiguracji przycisków, które mają zostać wykorzystane. Ich parametryzację opisano w poprzednim podrozdziale.

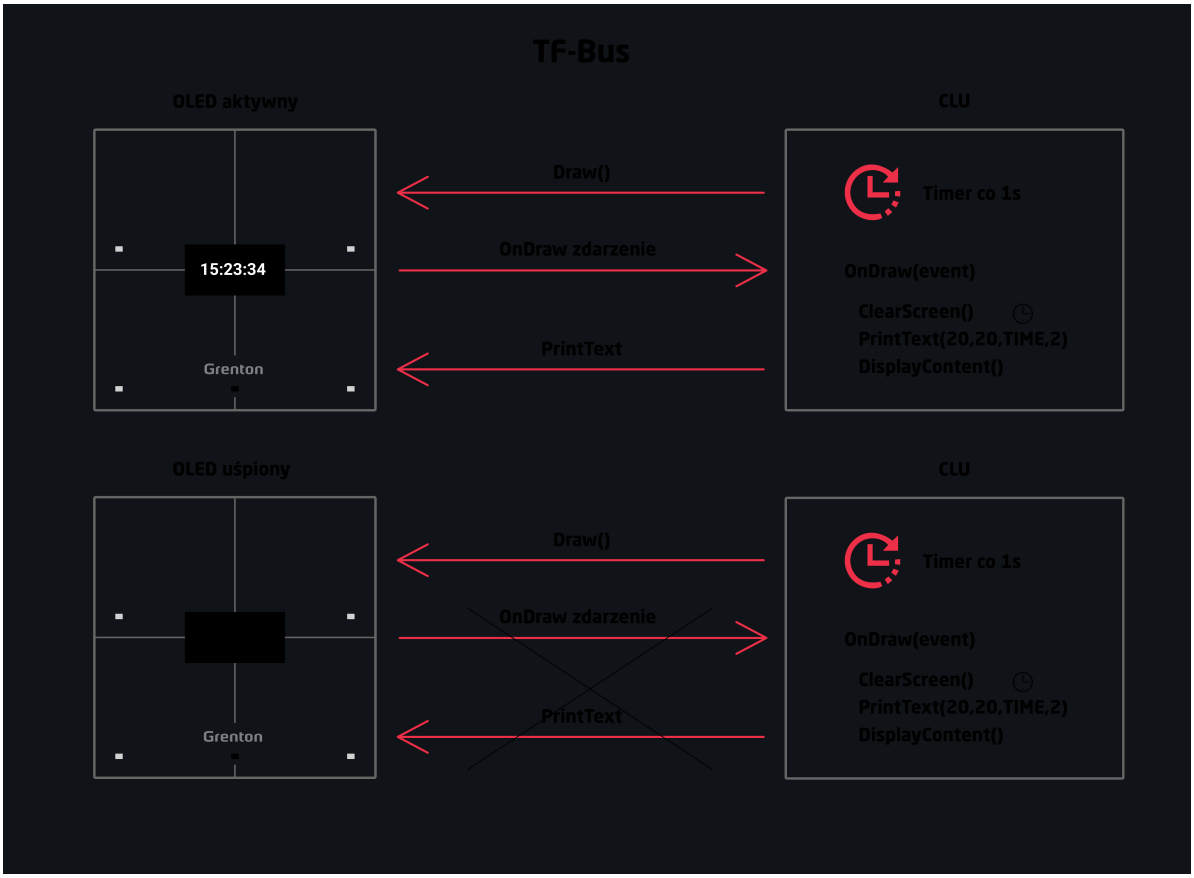
Następnym krokiem w tworzeniu konfiguracja powinno być stworzenie skryptów rysujących treść na wyświetlaczu modułu Smart Panel. Ich tworzenie jest analogiczne jak w wersji v3 modułu Smart Panel (patrz rozdział XII.4).

Przykład skryptu rysującego treść na wyświetlaczu (*Strona1*):

```
CLU220000260->x250000053_PANEL1->ClearScreen()
CLU220000260->x250000053_PANEL1->PrintText(15,10,"Kuchnia [°C]:",2)
CLU220000260->x250000053_PANEL1->PrintFloat(80,38,CLU220000260-
>x240000659_PANELSENSTEMP1->Value,1,2)
CLU220000260->x250000053_PANEL1->DrawLine(0,32,127,32,1)
CLU220000260->x250000053_PANEL1->DrawPoint(0,0,1)
CLU220000260->x250000053_PANEL1->DrawLine(70,32,70,63,1)
CLU220000260->x250000053_PANEL1->PrintText(15,40,CLU220000260->Time,1)
CLU220000260->x250000053_PANEL1->DisplayContent()
```

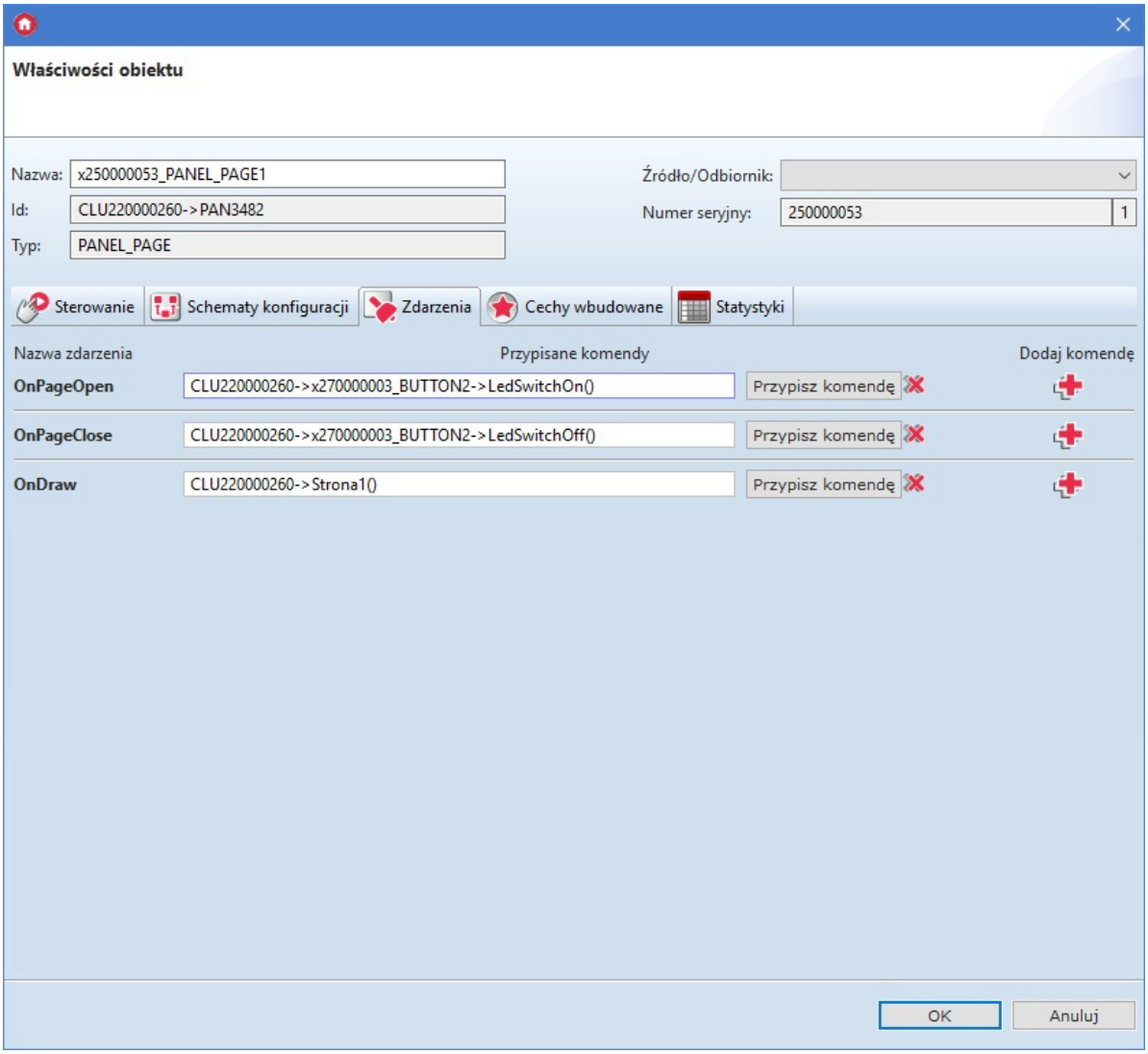
> **Uwaga!** W mechanizmie rysowania wprowadzone została ograniczenie. CLU Z-Wave oczekuje 2 sekundy na zakończenie rysowania metodą `DisplayContent`. W przeciwnym wypadku na ekranie wyświetlony zostanie komunikat o następującej treści:
>
> "page: `PageName`
>
> free draw
>
> ! TIMEOUT !"

Poniższy rysunek przedstawia obecny mechanizm rysowania.



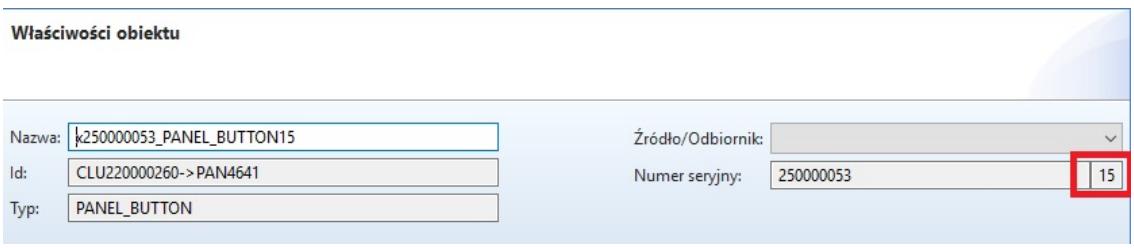
Kolejnym krokiem w tworzeniu konfiguracji jest konfiguracja obiektów Panel_Page w zależności od ilości przycisków. Jeden obiekt Panel_Page obsługuje do 4 przycisków. W tym celu należy:

- Otworzyć obiekt *PANEL_PAGEX* (gdzie X to numer kolejnej strony) poprzez dwukrotne kliknięcie na liście modułów;
- Przejście do zakładki *Zdarzenia*;
- Skonfigurować działanie strony poprzez przypisanie metod do określonych zdarzeń (klikając w "+" po prawej stronie okna):

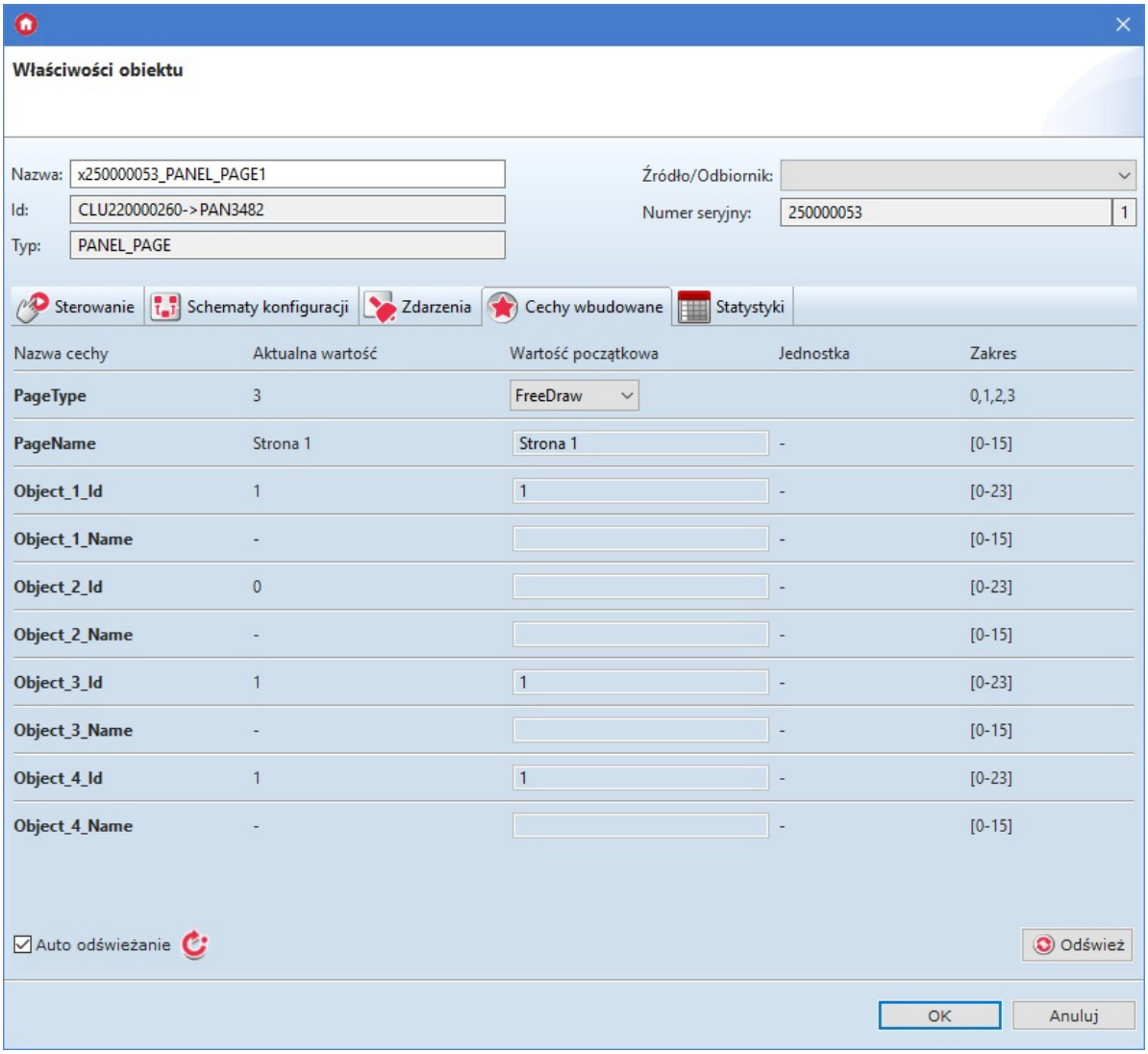


Uwaga! W przypadku typu strony *FreeDraw* należy uzupełnić zdarzenie `OnDraw`.

- Wybrać zakładkę *Cechy wbudowane* i zdefiniować obsługiwany typ strony i powiązać obiekty strony z przyciskami:
  - `PageType` - cecha określająca typ strony, należy ustawić ją na wartość *FreeDraw (3)*;
  - `PageName` - cecha określająca nazwę strony lub ikonę, która będzie wyświetlana w momencie przechodzenia pomiędzy stronami (działa wyłącznie przy ustawionej cesze `PageDisplayMode` na wartość 1 (ShowIconOrName) w obiekcie Panel);
  - `Object_X_Id` - identyfikator/numer przycisku. W tym celu należy odczytać wartość znajdującą się w polu *Numer seryjny* obiektu *PANEL_BUTTONX*



  - `Object_X_Name` - nazwa termostatu. W przypadku typu strony *FreeDraw* cechę należy pozostawić pustą;



Uwaga! Wysłanie konfiguracji wyłącznie ze zdefiniowanym typem strony, bez ustawionych powiązań obiektów z przyciskami wiąże się z uruchomieniem trybu pracy panelu jako *FreeDraw*. Jednakże przyciski na module będą nieaktywne. Ma to związek z nie uzupełnieniem cech Object_X_Id.

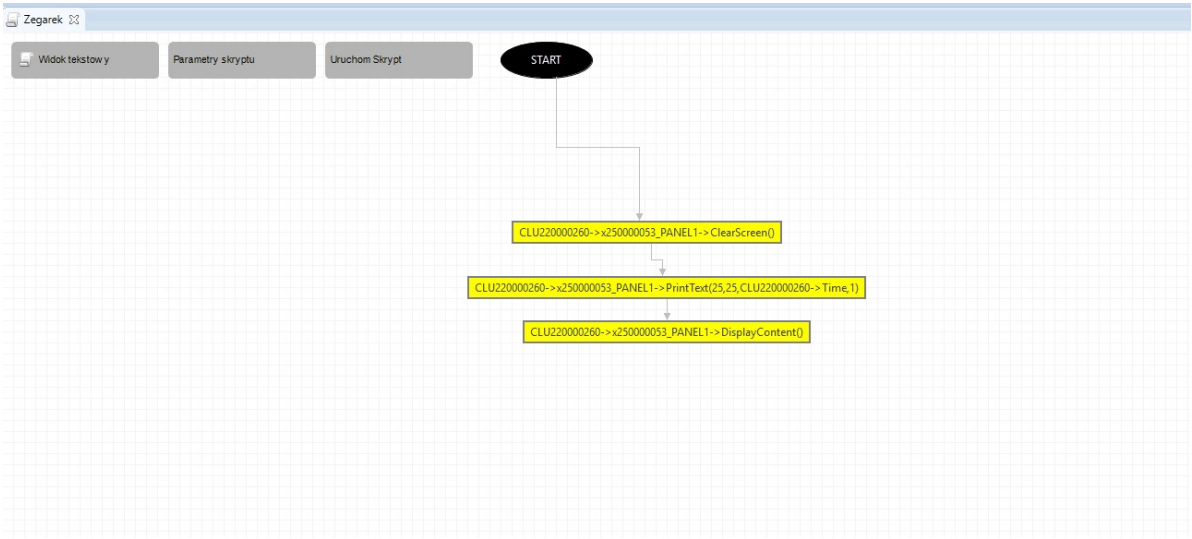
- Wysłać konfigurację do CLU Z-Wave.

Uwaga! Istnieje możliwość nadpisania treści wyświetlacza poprzez wywołanie metod rysujących z poziomu aplikacji Object Manager lub za pośrednictwem innych skryptów, które nie są przypisane do zdarzenia `OnDraw`. Jednakże, nadpisana treść zostanie wyczyszczona w momencie przejścia do innej strony lub wywołaniu metody `Draw` i wybudzeniu ekranu.

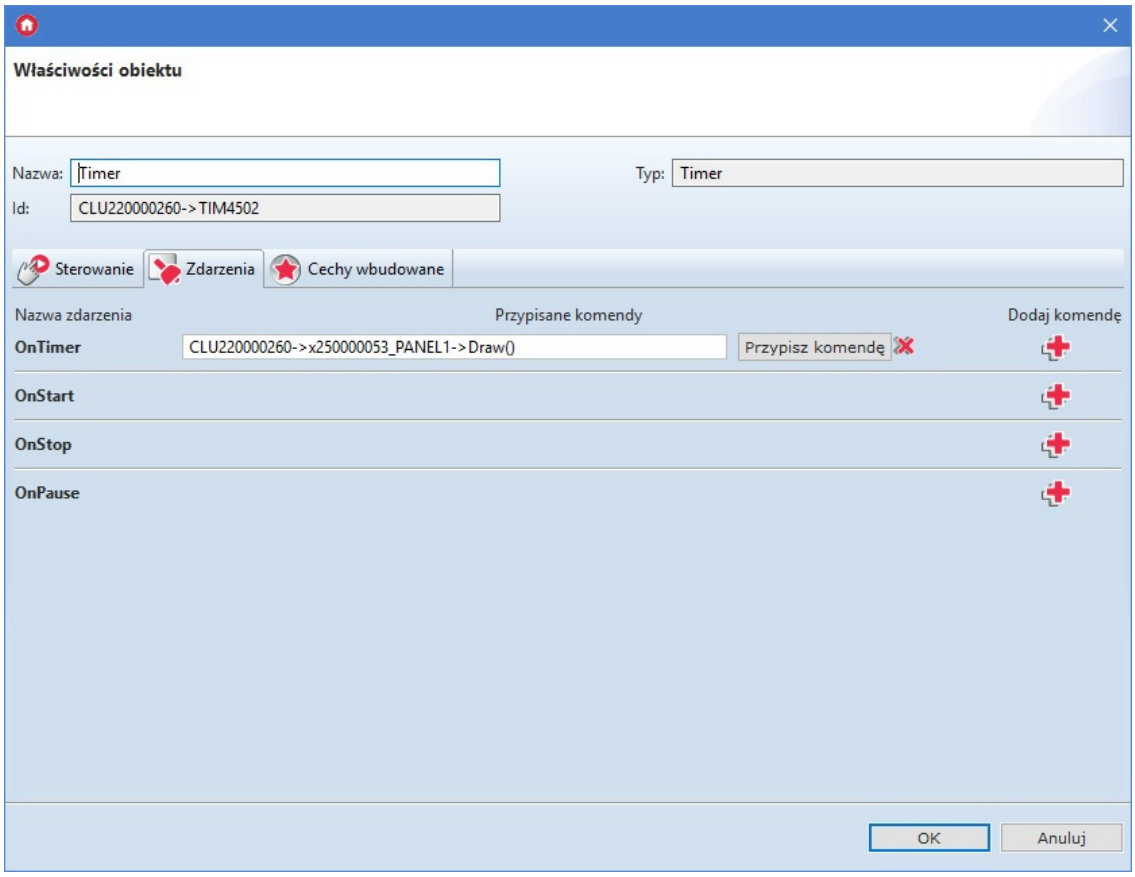
B. Konfiguracja strony jako zegarka

Chcąc skonfigurować stronę jako zegarek należy:

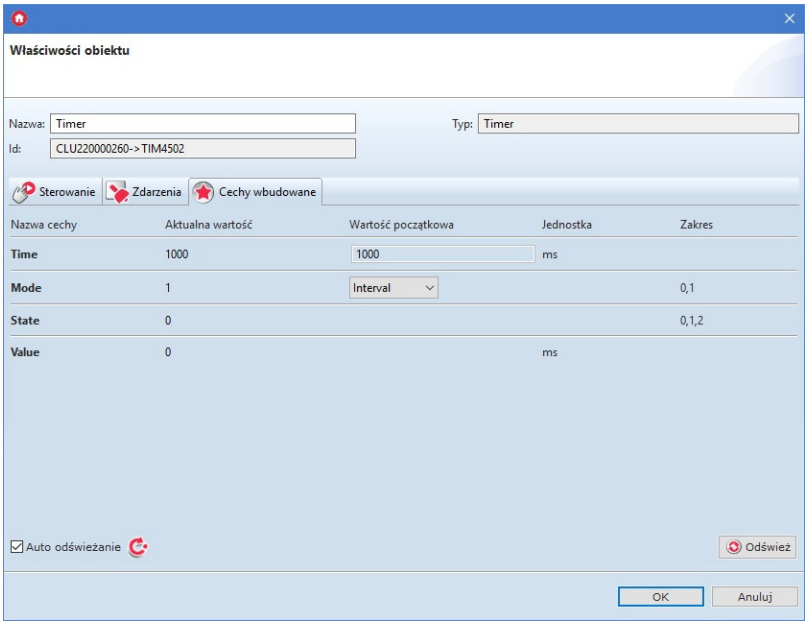
- Stworzyć skrypt wyświetlający aktualny czas (*Zegarek*);



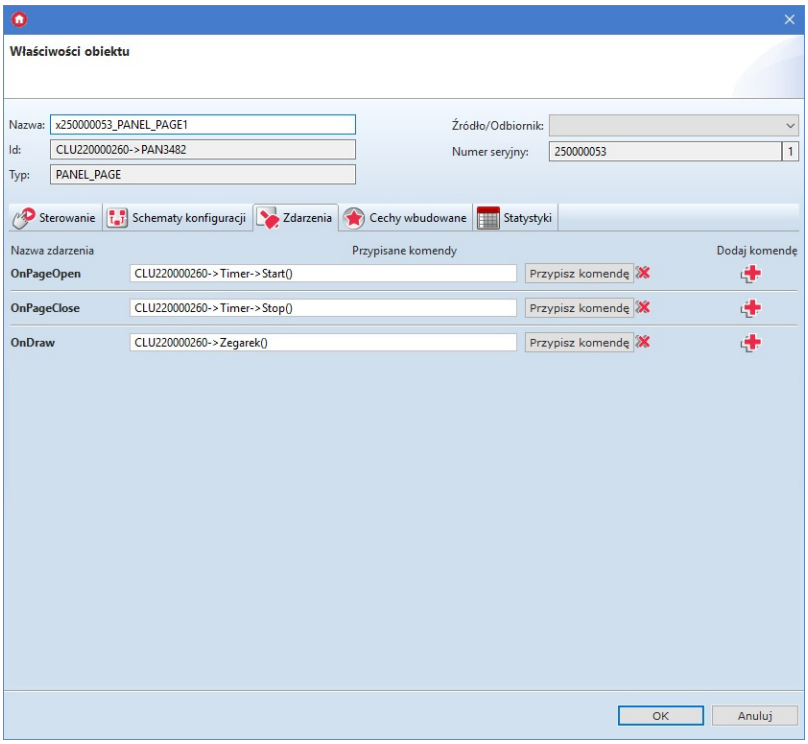
- Stworzyć obiekt wirtualny Timer:
  - Przejść do zakładki *Zdarzenia*;
  - Skonfigurować działanie obiektu wirtualnego poprzez przypisanie metody `Draw` obiektu *Panel* do zdarzenia `OnTimer`:



  - Wybrać zakładkę *Cechy wbudowane* i zdefiniować parametry konfiguracyjne obiektu:



- Otworzyć obiekt *PANEL_PAGEX* (gdzie X to numer strony) poprzez dwukrotne kliknięcie na liście obiektów:
    - Przejść do zakładki *Zdarzenia*
    - Skonfigurować działanie strony poprzez przypisanie metod do określonych zdarzeń (klikając w "+" po prawej stronie okna):



    - Wybrać zakładkę *Cechy wbudowane* i zdefiniować parametry konfiguracyjne obiektu;
- Wysłać konfigurację do CLU Z-Wave.

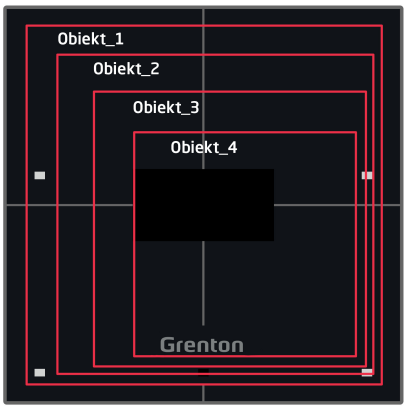
Skrypt *Zegarek* w wersji tekstowej:

```
CLU220000260->x250000053_PANEL1->ClearScreen()
CLU220000260->x250000053_PANEL1->PrintText(25,25,CLU220000260->Time,1)
CLU220000260->x250000053_PANEL1->DisplayContent()
```

5.9. Tworzenie konfiguracji z wykorzystaniem obiektu strony Thermostats

W trybie pracy *Thermostats* do dyspozycji jest strona składająca się z 4 obiektów (łącznie obsługa do 16 obiektów na 4 stronach), dla których przypisywane są obiekty termostatów zdefiniowanych w systemie. Możliwa jest zmiana parametrów termostatów takich jak temperatura zadana czy tryb pracy. Istnieje również możliwość włączenia lub wyłączenia danego termostatu.

> **Uwaga!** W trybie pracy *Thermostats* przyciski, jak również rysowanie treści na wyświetlaczu jest zablokowane.



Strona typu „Thermostats"

Tworzenie konfiguracji panelu obsługującej stronę lub strony typu *Thermostats* najlepiej rozpocząć od stworzenia termostatów, które mają zostać wykorzystane w konfiguracji. Opis tworzenia oraz pracy obiektu wirtualnego *Thermostat* opisany został w podrozdziale IX.5.

Wersja v4 modułu Smart Panel obsługuje dwa rodzaje termostatów:

- Termostat lokalny - jest to obiekt wirtualny typu *Thermostat* stworzony na module CLU Z-Wave, do którego podłączony jest modułu Smart Panel z tworzoną aktualnie konfiguracją;
- Termostat zdalny - jest to obiekt wirtualny typu *Thermostat* stworzony na innym module CLU Z-Wave;

Za pośrednictwem modułu Smart Panel istnieje możliwość zmiany takich parametrów obiektu wirtualnego *Thermostat* jak:

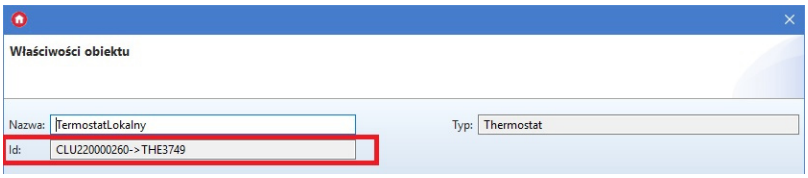
- `PointValue` - temperatura zadana, możliwość odczytu aktualnie ustawionej temperatury, jak również zmiany na nową wartość;
- `Mode` - tryb pracy termostatu:
  - W trybie automatycznym `Auto (2)` wartość temperatury odczytywana jest z harmonogramu. Za pośrednictwem modułu Smart Panel nie ma możliwości zmiany tej temperatury;
  - W trybie ręcznym `Manual (0)` wartość temperatury odczytywana jest z cechy `PointValue`. Za pośrednictwem modułu Smart Panel istnieje możliwość zmiany tej temperatury;

- `State` - aktualny stan termostatu: wyłączony (`Off (0)`) / włączony (`On (1)`).

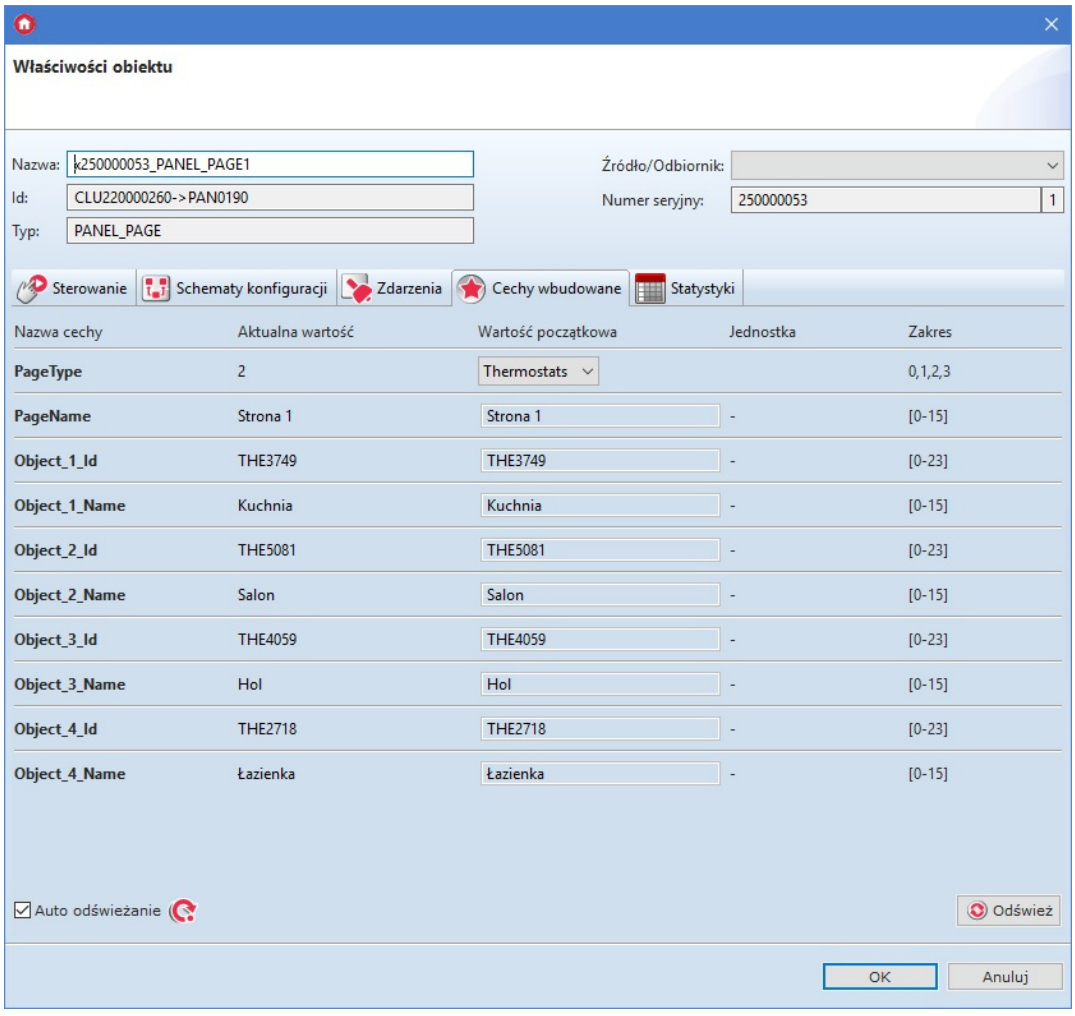
A. Tworzenie konfiguracji z termostatem lokalnym

Chcąc stworzyć konfigurację z wykorzystaniem termostatu lokalnego należy:

- Stworzyć termostat na CLU Z-Wave, do którego podłączony jest moduł Smart Panel;
- Skonfigurować obiekt wirtualny zgodnie z założeniami;
- Otworzyć obiekt *PANEL_PAGEX* (gdzie X to numer jednej z 4 stron) poprzez dwukrotne kliknięcie na liście obiektów
- Wybrać zakładkę *Cechy wbudowane* i zdefiniować obiekty wyświetlane na ekranie:
  - `PageType` - cecha określająca typ strony, należy ustawić ją na wartość *Thermostats (2)*;
  - `PageName` - cecha określająca nazwę strony lub ikonę, która będzie wyświetlana w momencie przechodzenia pomiędzy stronami (działa wyłącznie przy ustawionej cesze `PageDisplayMode` na wartość 1 (ShowIconOrName) w obiekcie Panel);
  - `Object_X_Id` - identyfikator termostatu. W tym celu należy odczytać wartość znajdującą się w polu *Id* obiektu wirtualnego *Thermostat*. Identyfikator termostatu lokalnego nie jest poprzedzony identyfikatorem CLU:



  - `Object_X_Name` - nazwa termostatu. Brak nazwy termostatu w parametrze powoduje, iż termostat nie jest wyświetlany;



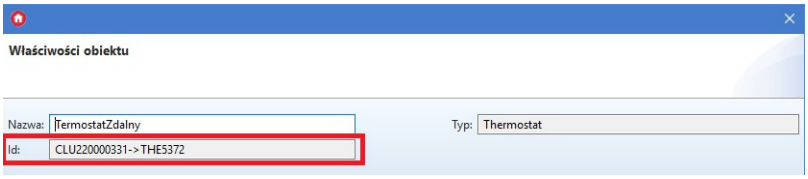
> **Uwaga!** Wysłanie konfiguracji wyłącznie ze zdefiniowanym typem strony, bez ustawionych powiązań obiektów z przyciskami wiąże się z uruchomieniem trybu pracy panelu jako *Thermostats*. Na wyświetlaczu będą widoczne kropki ("..."). Ma to związek z nie uzupełnieniem cech `Object_X_Id` oraz `Object_X_Name`.

- Wysłać konfigurację do CLU Z-Wave.

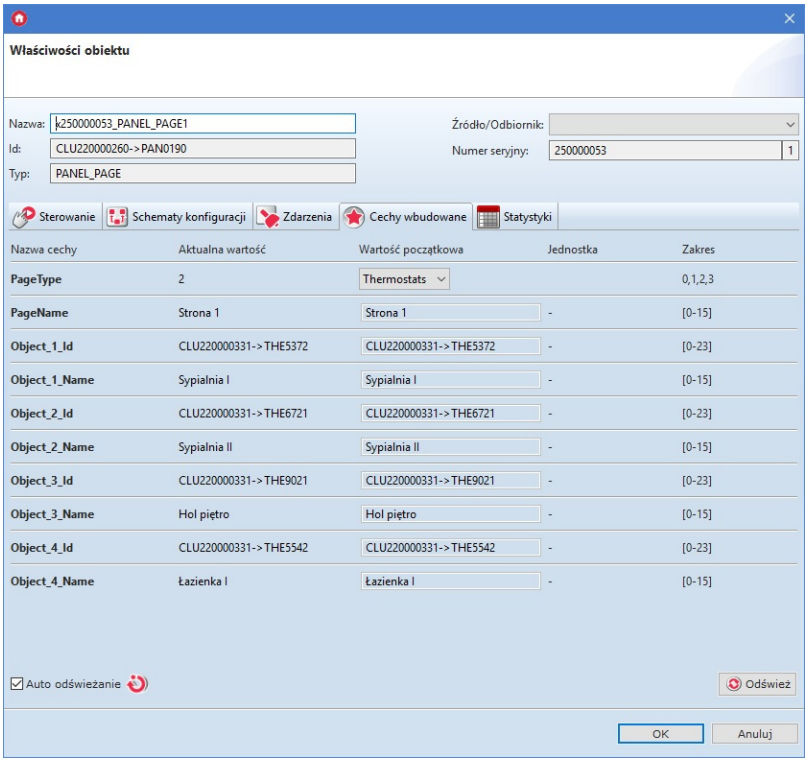
B. Tworzenie konfiguracji z termostatem zdalnym

Chcąc stworzyć konfigurację z wykorzystaniem termostatu zdalnego należy:

- Stworzyć termostat na CLU Z-Wave, do którego nie podłączony jest moduł Smart Panel z aktualnie wykonywaną konfiguracją;
- Skonfigurować obiekt wirtualny zgodnie z założeniami;
- Otworzyć obiekt *PANEL_PAGEX* (gdzie X to numer jednej z 4 stron) poprzez dwukrotne kliknięcie na liście obiektów
- Wybrać zakładkę *Cechy wbudowane* i zdefiniować obiekty wyświetlane na ekranie:
  - `PageType` - cecha określająca typ strony, należy ustawić ją na wartość *Thermostats (2)*;
  - `PageName` - cecha określająca nazwę strony lub ikonę, która będzie wyświetlana w momencie przechodzenia pomiędzy stronami (działa wyłącznie przy ustawionej cesze `PageDisplayMode` na wartość 1 (ShowIconOrName) w obiekcie Panel);
  - `Object_X_Id` - identyfikator termostatu. W tym celu należy odczytać wartość znajdującą się w polu *Id* obiektu wirtualnego *Thermostat*. Identyfikator termostatu zdalnego musi zostać poprzedzony identyfikatorem CLU:



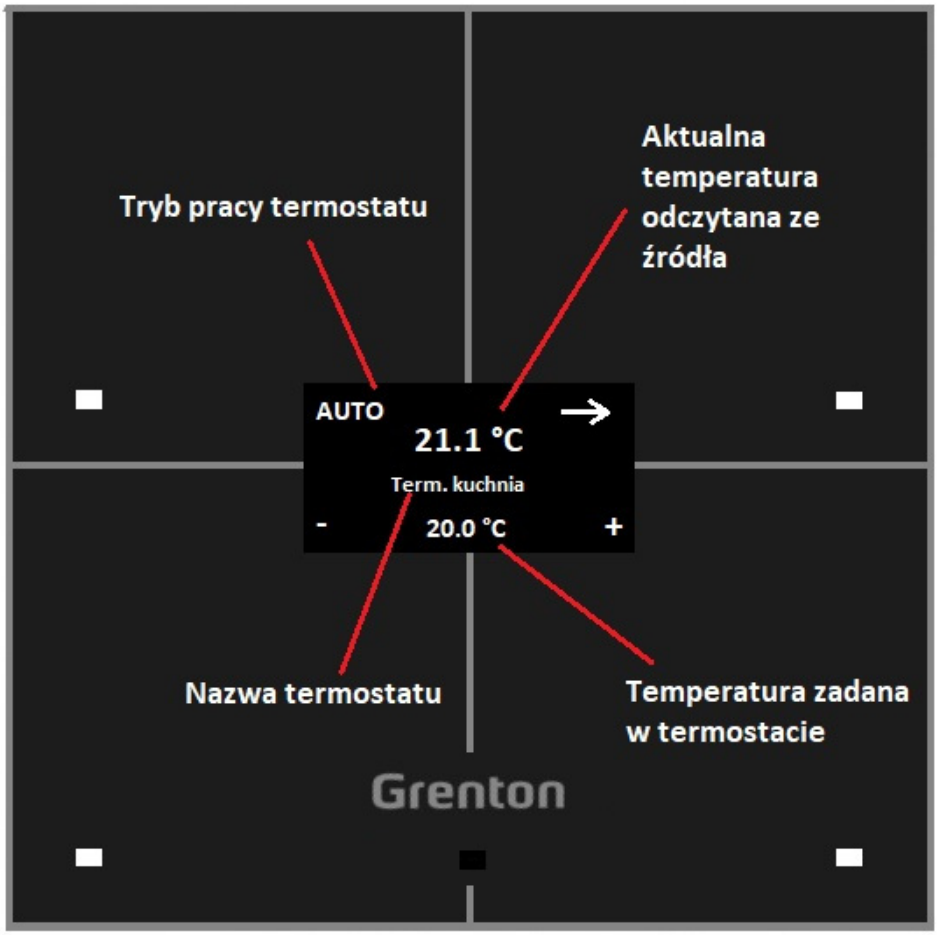
  - `Object_X_Name` - nazwa termostatu. Brak nazwy termostatu w parametrze powoduje, iż termostat nie jest wyświetlany;



> **Uwaga!** Wysłanie konfiguracji wyłącznie ze zdefiniowanym typem strony, bez ustawionych powiązań obiektów z przyciskami wiąże się z uruchomieniem trybu pracy panelu jako *Thermostats*. Na wyświetlaczu będą widoczne kropki ("..."). Ma to związek z nie uzupełnieniem cech `Object_X_Id` oraz `Object_X_Name`.

- Wysłać konfigurację do CLU Z-Wave.

Poniższy schemat przedstawia poglądowy wygląd termostatu na ekranie Smart Panelu. Za pośrednictwem strzałki użytkownik może przejść do kolejnego termostatu na stronie. Natomiast za pomocą "-" / "+" istnieje możliwość zmiany zadanej temperatury.



C. Predefiniowane zachowanie przycisków

Przycisk	Krótkie/długie przyciśnięcie	Opis zachowania
Lewy górny	Krótkie naciśnięcie (kliknięcie)	Zmiana trybu pracy termostatu: Manual/Auto
Lewy górny	Długie naciśnięcie (przytrzymanie)	Zmiana stanu termostatu: Off/On
Prawy górny	Krótkie naciśnięcie (kliknięcie)	Przejście do następnego termostatu na stronie
Prawy górny	Długie naciśnięcie (przytrzymanie)	Brak zdefiniowanej funkcjonalności
Lewy dolny	Krótkie naciśnięcie (kliknięcie)	Zmniejszenie temperatury zadanej (`PointValue`) o 0.1°C
Lewy dolny	Długie naciśnięcie (przytrzymanie)	Zmniejszenie temperatury zadanej (`PointValue`) - dopóki przycisk jest przytrzymany
Prawy dolny	Krótkie naciśnięcie (kliknięcie)	Zwiększenie temperatury zadanej (`PointValue`) o 0.1 °C
Prawy dolny	Długie naciśnięcie (przytrzymanie)	Zwiększenie temperatury zadanej (`PointValue`) - dopóki przycisk jest przytrzymany

5.10. Łączenie obiektów w większe przyciski

Nowa wersja modułu Smart Panel wprowadza również możliwość łączenia/scalania 2, 3 lub 4 obiektów w jeden większy przycisk. Funkcjonalność dostępna jest wyłącznie w trybie stron *Buttons* oraz *FreeDraw*. W celu stworzenia większego przycisku należy:

- Skonfigurować obiekty *PANEL_BUTTOX* (gdzie X to numer przycisku):
  - W zakładce *Zdarzenia* skonfigurować działanie przycisku poprzez przypisanie metod do określonych zdarzeń;
  - W zakładce *Cechy wbudowane* zdefiniować obiekty wyświetlane na ekranie danego przycisku;
- Otworzyć obiekt *PANEL_PAGEX* (gdzie X to numer strony);
- Przejść do zakładki *Zdarzenia*;
- Skonfigurować działanie strony poprzez przypisanie metod do określonych zdarzeń;
- Przejść do zakładki *Cechy wbudowane*;
- Ustawić cechę `PageType` na wartość *Buttons* lub *FreeDraw*;
- Ustawić cechy `Object_X_Id` zgodnie z żądaną wersją łączenia:
  - Scalanie 2 obiektów w jeden przycisk w poziomie - ikona ustawiona dla przycisku jest wyświetlana na środku w górnej części ekranu (dla obiektów `Object_1_Id` oraz `Object_2_Id`) lub dolnej części ekranu (dla obiektów `Object_3_Id` oraz `Object_4_Id`);
  - Scalanie 2 obiektów w jeden przycisk w pionie - ikona ustawiona dla przycisku jest wyświetlana na środku po lewej części ekranu (dla obiektów `Object_1_Id` oraz `Object_3_Id`) lub po prawej części ekranu (dla obiektów `Object_2_Id` oraz `Object_4_Id`);

- Scalanie 3 obiektów w jeden przycisk - wyświetlana są dwie identyczne ikony, w zależności od sposobu łączenia obiektów;
- Scalanie 4 obiektów w jeden przycisk - ikona ustawiona dla przycisku jest wyświetlana na środku ekranu

6. Konfiguracja modułu Smart Panel w wersji v6

> **Uwaga!** Smart Panel w wersji v6 dostępny jest dla Object Managera w wersji 1.4.1 i wyższej oraz dla CLU Z-Wave z firmware 5.08.01 i wyższym.

6.1. Parametry konfiguracyjne

W najnowszej wersji modułu Smart Panel v6 zostały wprowadzone nowe parametry konfiguracyjne dla takich obiektów jak:

- PANEL,
- PANEL_PAGE,
- PANELSENSTEMP.

Pełną listę zmian wprowadzonych w wersji V6 można znaleźć w release notes danej wersji: Release Notes - moduł Smart Panel

6.2. Nowa funkcjonalność

A. Mechanizm informowania o błędnej konfiguracji / przejściu w tryb Rozproszonej Logiki

Nową funkcjonalnością wprowadzoną wraz z wersją v6 jest mechanizm informowania użytkownika o błędnej konfiguracji lub przejścia w tryb Rozproszonej Logiki. Mechanizm ten polega na tym, iż moduł oczekuje około 10 sekund na otrzymanie konfiguracji po jej wysłaniu lub restarcie systemu. Po tym czasie kończy się okres oczekiwania na konfigurację, a użytkownik zostanie poinformowany o błędnej konfiguracji lub przejściu w tryb Rozproszonej Logiki poprzez jeden, krótki i niski dźwięk.

B. Tryb Rozproszonej Logiki

Kolejną funkcjonalnością dodaną do najnowszej wersji modułu Smart Panel jest tryb Rozproszonej Logiki. Jest on dostępny od wersji 6.1.8-2115 i wyższej. Szczegółowe informacje na temat konfiguracji oraz sposobu działania Rozproszonej Logiki - patrz XIX.

6.3. Zmiana UI oraz mechanizmu działania stron typu Thermostats

Moduł Smart Panel w wersji v6 oferuje odświeżony UI strony typu Thermostats, a także nowe funkcje. Sposób tworzenia i konfiguracji strony z wykorzystaniem obiektów typu Thermostats jest taki sam jak dla poprzednich wersji - patrz - XII.5.9.

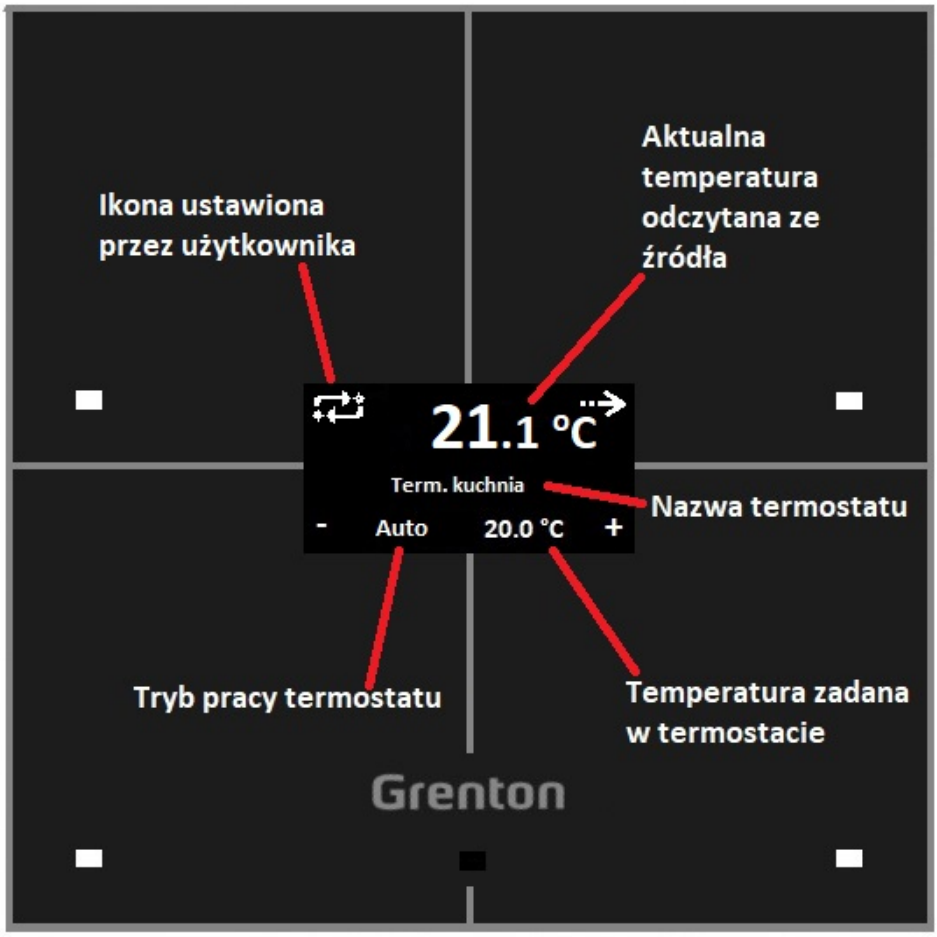
A. Zmiana UI termostatu

Poniższy schemat przedstawia poglądowy wygląd odświeżonego termostatu na ekranie Smart Panelu. Zmianie uległo kilka elementów:

- wprowadzenie ikony w lewym górnym rogu - domyślnie wyświetlana jest ikona "chmode", która znajduje się na karcie SD (ikona została przedstawiona na schemacie). Jednakże w przypadku braku niniejszej ikony na karcie SD, zostanie wyświetlony napis "mode". Dodatkowo użytkownik może

wprowadzić własną ikonę za pośrednictwem metody `SetObject_X_CustomIcon` lub też cechy wbudowanej `Object_X_CustomIcon`,

- za pośrednictwem strzałki użytkownik może przejść do kolejnego termostatu na stronie (krótkie naciśniecie przycisku w przypadku więcej niż jednego termostatu na stronie) lub przejść do następnej strony (krótkie naciśnięcie w przypadku jednego termostatu na stronie, dłuższe przytrzymanie przycisku w przypadku więcej niż jednego termostatu na stronie),
- przy strzałce zostały wprowadzone kropki, które sygnalizują numer termostatu obecnie wyświetlanego (jedna kropka - Object_1_Id, dwie kropki - Object_2_Id, itd.). W przypadku kiedy na stronie jest tylko i wyłącznie jeden termostat, kropki nie są wyświetlane,
- za pomocą "-" / "+" istnieje możliwość zmiany zadanej temperatury oraz trybu działania termostatu z Auto na Manual,
- dłuższe przytrzymanie lewego górnego przycisku (w dalszej części zwanego 'mode') powoduje wyłączenie / włączenie termostatu lub zmianę z trybu Manual na tryb Auto,
- w przypadku kiedy termostat jest wyłączony, to znika zadana temperatura, a pojawia się napis "Off", który usytuowany jest centralnie,
- zmianie uległo również wyświetlanie temperatury odczytanej ze źródła - obecnie większą czcionką wyświetlany jest wartość temperatury przed przecinkiem, natomiast mniejszą wartość po przecinku oraz jednostka. Dodatkowo wyświetlanie tej temperatury jest zależne od wielkości wprowadzonej ikony - więcej na ten temat w kolejnym podpunkcie.



B. Nowe funkcje na stronie typu Thermostats

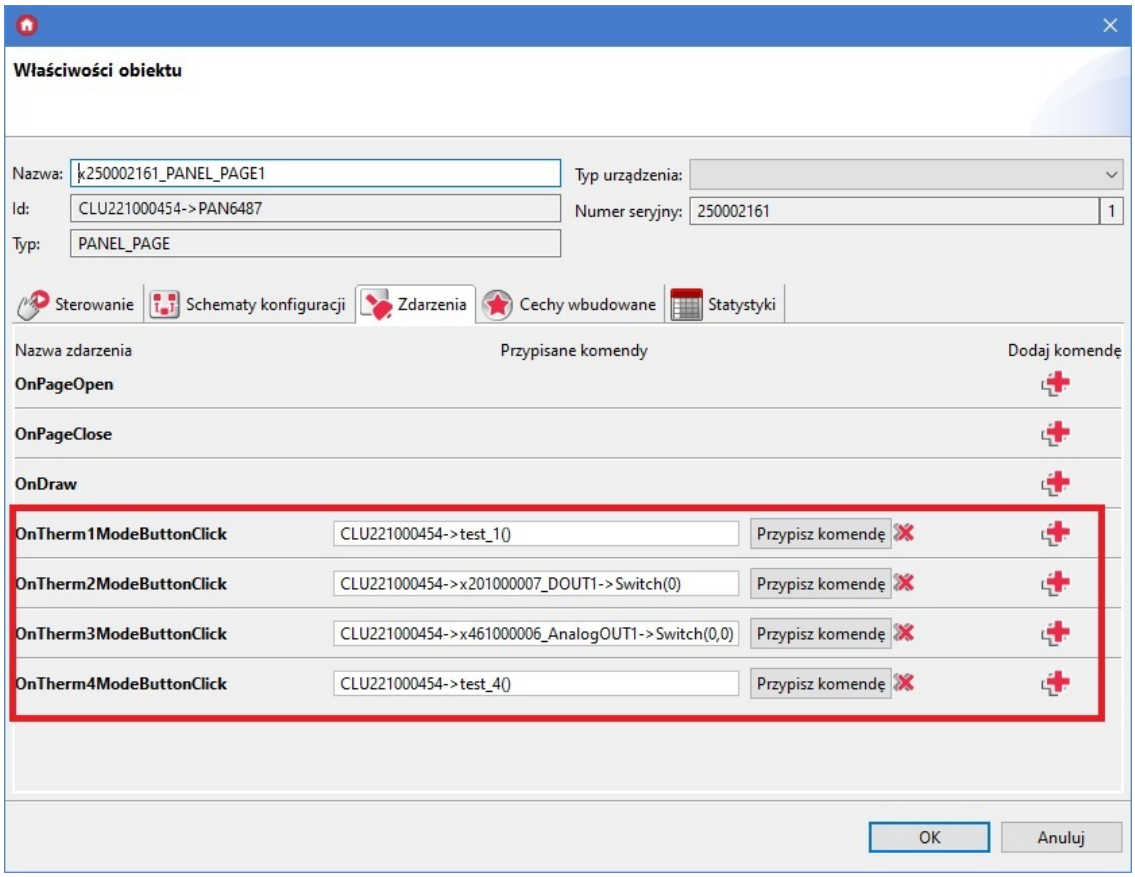
Od wersji v6 modułu Smart Panel dostępne są nowe funkcjonalności na stronie typu Thermostats. Jest to związane z wprowadzonymi zmianami w obrębie UI termostatu.

1. **Możliwość ustawiania własnych ikon**

Pierwszą nową funkcjonalnością jest wspomniana wcześniej możliwość ustawiania własnych ikon przez użytkownika w lewym górnym narożniku wyświetlacza. Ich zmiana jest możliwa zarówno poprzez metodę `SetObject_X_CustomIcon` jak również poprzez cechę wbudowaną `Object_X_CustomIcon`. Szerokość wprowadzonej ikony ma wpływ na współrzędną "x" aktualnej temperatury. Dopuszczalne są ikony o wymiarach 64 x 32 piksele. W przypadku przekroczenia szerokości ikony x > 64, na wyświetlaczu zostanie narysowana wyłącznie sama ikona - UI termostatu nie będzie wyświetlany - jest to tak zwany tryb "big icon". Aby powrócić do wyświetlania interfejsu związanego z termostatem, należy ustawić ikonę, której szerokość nie przekracza 64 pikseli.

2. **Możliwość przypisania akcji pod nowe zdarzenia**

   Następną funkcjonalnością wprowadzoną w nowej wersji modułu jest możliwość przypisywania akcji pod nowe zdarzenia `OnThermXModeButtonClick`, gdzie X oznacza numer termostatu na stronie. Zdarzenie to jest generowane w momencie kliknięcia w przycisk 'mode' (lewy, górny przycisk).



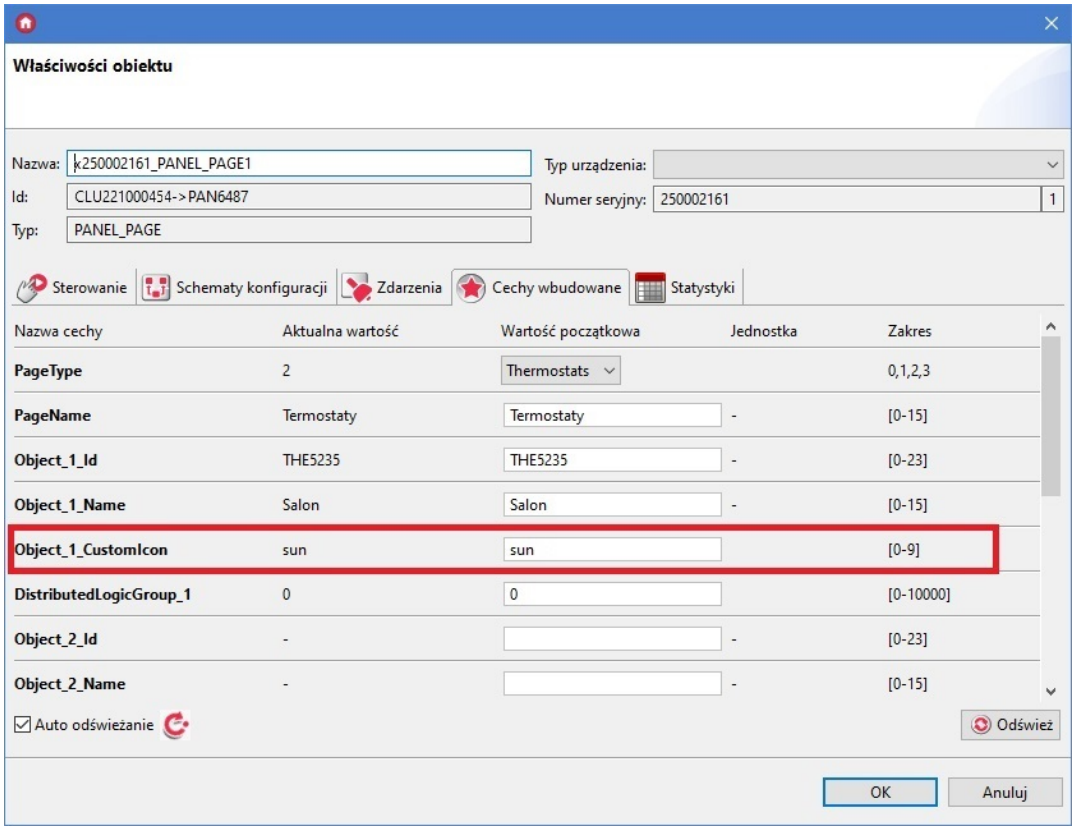
3. **Przykład konfiguracji nowych funkcjonalności**

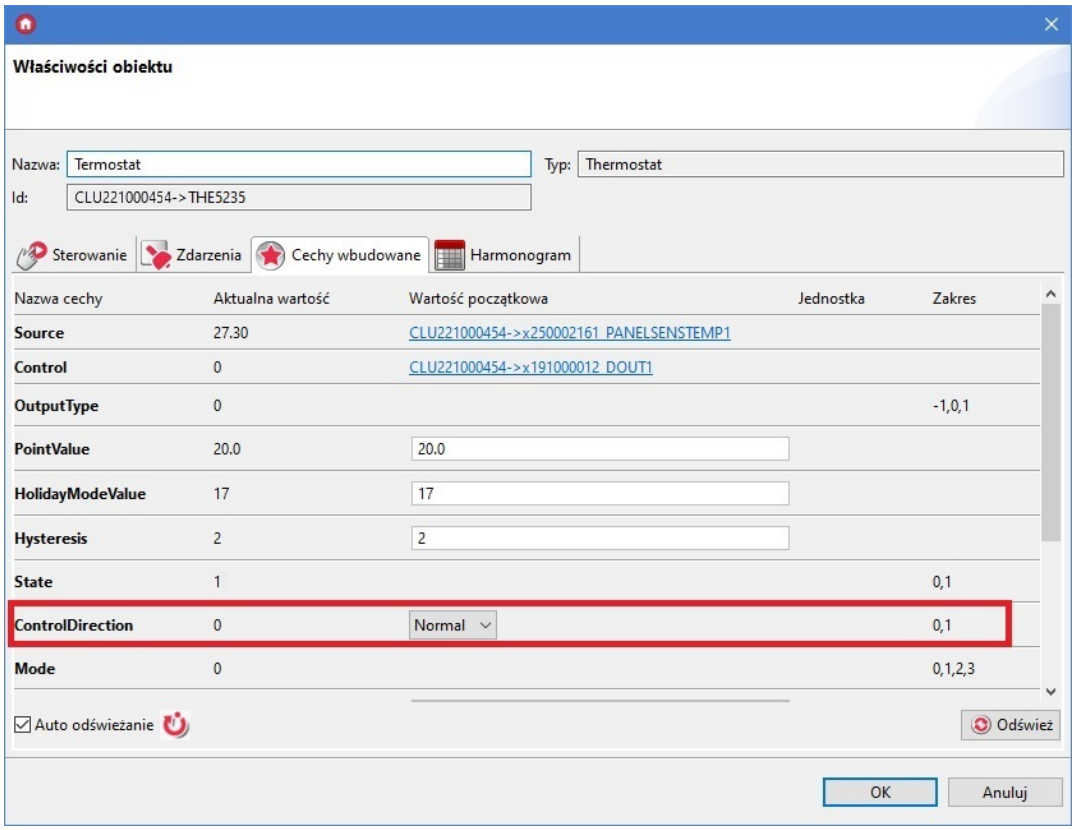
   1. Konfiguracja lokalnego termostatu wraz ze zmianą pracy (grzanie / chłodzenie)

      Do stworzenia niniejszej konfiguracji wykorzystane są takie obiekty jak:

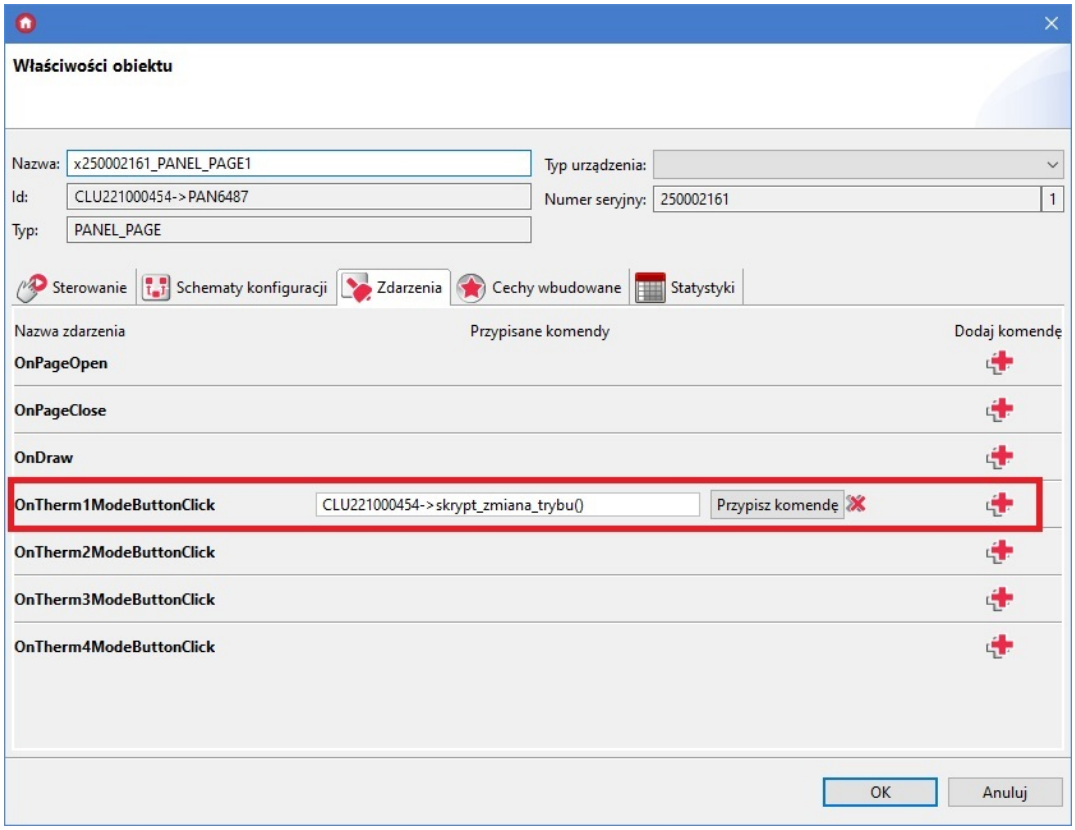
      - obiekt wirtualny Thermostat,
      - obiekt DOUT1 (np. modułu Relay) - odpowiedzialny za załączanie grzania / chłodzenia - wykorzystany w obiekcie wirtualnym Thermostat,
      - obiekt DOUT2 (np. modułu Relay) - selekcja co ma zostać załączone: grzanie czy chłodzenie - wykorzystany w skryptach.

      Na poniższym zrzucie ekranu przedstawiona została konfiguracja obiektu PANEL_PAGE oraz obiektu wirtualnego Thermostat w domyślnym ustawieniu, czyli grzaniu.

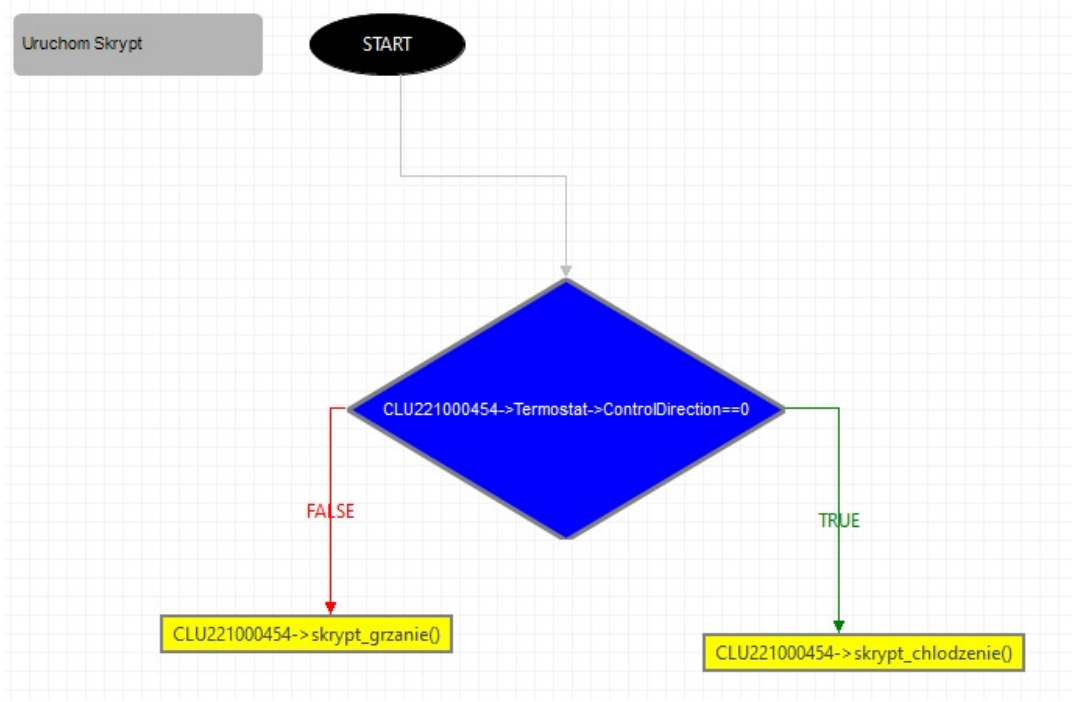




Konfiguracja zdarzenia OnTherm1ModeButtonClick:



Skrypt *skrypt_zmiana_trybu()* dokonujący zmiany trybu pracy termostatu z grzania na chłodzenie i na odwrót:



Powyższy skrypt w wersji tekstowej:

```
if(CLU221000454->Termostat->ControlDirection==0) then
CLU221000454->skrypt_chlodzenie()
else
CLU221000454->skrypt_grzanie()
end
```

Skrypt *skrypt_grzanie()* służący do zmiany cechy ControlDirection na wartość odpowiedzialną za grzanie, a także zmianę ikony i selekcję grzanie/chłodzenie:

```
CLU221000454->Termostat->SetControlDirection(0)
CLU221000454->x201000007_DOUT2->SwitchOff(0)
CLU221000454->x250002161_PANEL_PAGE1->SetObject_1_CustomIcon("sun")
```



Skrypt *skrypt_chlodzenie()* służący do zmiany cechy ControlDirection na wartość odpowiedzialną za chłodzenie, a także zmianę ikony i selekcję grzanie/chłodzenie:

```
CLU221000454->Termostat->SetControlDirection(1)
CLU221000454->x201000007_DOUT2->SwitchOff(1)
CLU221000454->x250002161_PANEL_PAGE1->SetObject_1_CustomIcon("cold")
```



C. Predefiniowane zachowanie przycisków

Przycisk	Krótkie/długie przyciśnięcie	Opis zachowania
Lewy górny	Krótkie naciśnięcie (kliknięcie)	Wygenerowanie zdarzenia OnThermXModeButtonClick, gdzie X to numer termostatu na stronie
Lewy górny	Długie naciśnięcie (przytrzymanie)	Zmiana stanu termostatu: Off/On. Dodatkowo umożliwia przejście z trybu Manual na tryb Auto
Prawy górny	Krótkie naciśnięcie (kliknięcie)	Przejście do następnego termostatu na stronie w przypadku więcej niż jednego termostatu na stronie Przejście do następnej strony w przypadku, kiedy na danej stronie jest tylko i wyłącznie jeden termostat
Prawy górny	Długie naciśnięcie (przytrzymanie)	Przejście do następnej strony
Lewy dolny	Krótkie naciśnięcie (kliknięcie)	Zmniejszenie temperatury zadanej (`PointValue`) o 0.1°C, a także zmiana trybu pracy z Auto na Manual
Lewy dolny	Długie naciśnięcie (przytrzymanie)	Zmniejszenie temperatury zadanej (`PointValue`) - dopóki przycisk jest przytrzymany, a także zmiana trybu pracy z Auto na Manual
Prawy dolny	Krótkie naciśnięcie (kliknięcie)	Zwiększenie temperatury zadanej (`PointValue`) o 0.1 °C, a także zmiana trybu pracy z Auto na Manual
Prawy dolny	Długie naciśnięcie (przytrzymanie)	Zwiększenie temperatury zadanej (`PointValue`) - dopóki przycisk jest przytrzymany, a także zmiana trybu pracy z Auto na Manual

XIII. Moduł GATE ALARM

> **Uwaga!** Opisana funkcjonalność oraz integracja z wymienionymi centralami alarmowymi jest dostępna dla **GRENTON GATE ALARM, DIN, Eth (INT-221-E-01)** posiadający **firmware 1.1.20-2219 lub wyższy!**

1. Informacje ogólne

Moduł GATE ALARM służy do integracji systemu Grenton Smart Home z urządzeniami i systemami zewnętrznymi. Umożliwia on integrację z systemem alarmowym Satel i Jablotron.

2. Konfiguracja modułu

> **Uwaga!** Przed rozpoczęciem jakiekolwiek pracy z modułem GATE Alarm wymagana jest aktualizacja bazy interfejsów!

Ustawianie czasu za pomocą serwera NTP

Moduł GATE Alarm umożliwia ustawianie czasu za pomocą serwera NTP wraz z uwzględnieniem strefy czasowej a także zmianą czasu (zimowy/letni). Czas pobierany jest automatycznie z serwera NTP (*pool.ntp.org*).

Do konfiguracji służą trzy cechy:

- `UseNTP` – określa czy GATE używa NTP,
- `NTPTimeout` - czas oczekiwania na odpowiedź z serwera NTP,
- `TimeZone` – ustawianie strefy czasowej GATE – dostępne są 22 strefy.

> **Uwaga!** Pobieranie czasu z serwera NTP wymaga, aby GATE znajdowało się w sieci, która posiada połączenie z internetem.

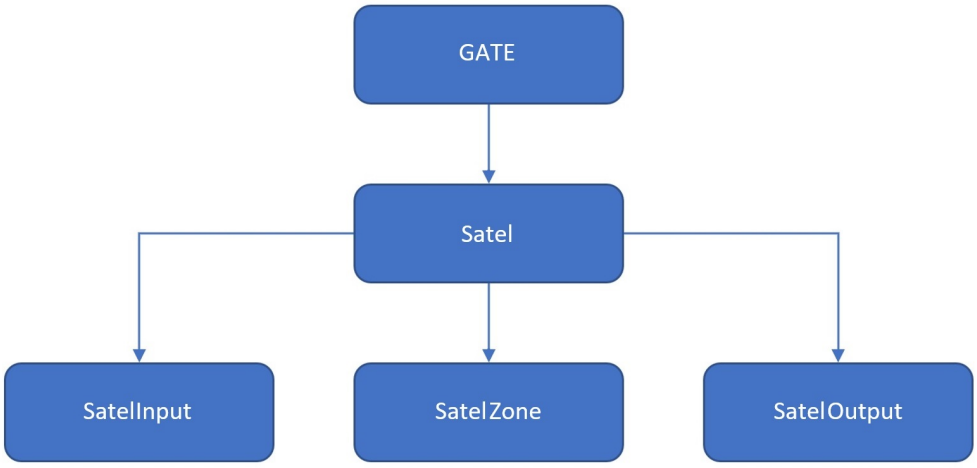
3. Integracja z centralą Satel

3.1. Informacje ogólne

Integracja systemu Grenton z centralą alarmową Satel jest możliwa za pośrednictwem modułu ETHM-1. Istnieje możliwość tworzenia obiektów wirtualnych typu: *SatelZone*, *SatelInput*, *SatelOutput*. Możliwe jest również wykorzystanie kodowania integracji oferowanego przez firmę Satel.

> **Uwaga!** Dla tworzonych obiektów wirtualnych nie ma ograniczenia względem ilości obiektów - ograniczeniem jest pamięć urządzenia, na którą wpływa m.in. poziom rozbudowania logiki na module. Wyjątkiem jest obiekt **Satel** - można stworzyć tylko jeden.

Konstrukcja konfiguracji wygląda następująco:



Obiekt wirtualny:

- **Satel** – umożliwia przeprowadzenie konfiguracji pozwalającej na integrację systemu z centralą alarmową Grenton;
- **SatelZone** – pozwala utworzyć strefę, do której dostęp będzie możliwy po wprowadzeniu hasła jednego z użytkowników lub hasła samego administratora;
- **SatelInput** – daje możliwość monitorowania stanu wybranego wejścia;
- **SatelOutput** – pozwala monitorować oraz ustawiać stan wybranego wyjścia po podaniu hasła użytkownika lub administratora.

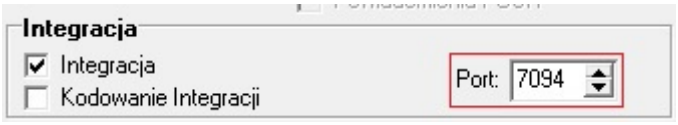
3.2. Konfiguracja dla systemu Satel

Uwaga! Przed rozpoczęciem jakiejkolwiek pracy z modułem GATE Alarm wymagana jest aktualizacja bazy interfejsów!

Uwaga! Wszystkie wymagane informacje można znaleźć w konfiguracji modułu ETHM – korzystając z manipulatora podłączonego do centrali Satel bądź za pomocą dedykowanego programu DLOADX.

Przed rozpoczęciem konfiguracji należy posiadać informacje dotyczące centrali Satel oraz modułu ETHM-1:

- Adres IP modułu ETHM (Satel) – dostępny w konfiguracji Satela *(DLOADX -> Dane -> Struktura i Sprzęt -> zakładka Sprzęt -> Manipulatory -> ETHM-1 -> sekcja Adres IP serwera)*;
- Port integracji ETHM – dostępny w konfiguracji Satela *(DLOADX -> Dane -> Struktura i Sprzęt -> zakładka Sprzęt -> Manipulatory -> ETHM-1 -> sekcja Integracja)*;



- Hasło administratora/użytkowników - domyślne hasło w konfiguracji Satela dla administratora to: 1111 *(DLOADX -> Użytkownicy -> Użytkownicy)*;

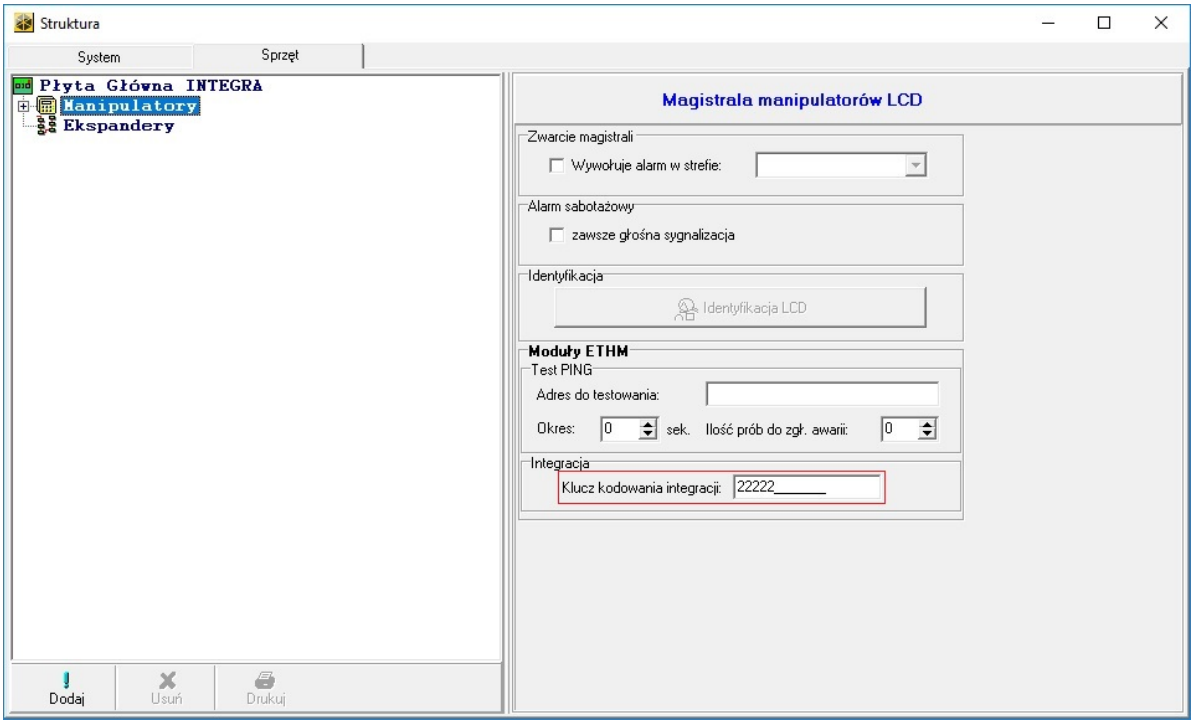
- Integracja po stronie modułu ETHM **musi być załączona** *(DLOADX -> Dane -> Struktura i Sprzęt -> zakładka Sprzęt -> Manipulatory -> ETHM-1 -> sekcja Integracja)*;



- W sytuacji, gdy szyfrowanie - *Kodowanie integracji* jest załączone, należy również znać klucz kodowania *(DLOADX -> Dane -> Struktura i Sprzęt -> zakładka Sprzęt -> Manipulatory -> ETHM-1 -> sekcja Integracja)*;



- Klucz kodowania można odnaleźć w konfiguracji Satela *(DLOADX -> Dane -> Struktura i Sprzęt -> zakładka Sprzęt -> Manipulatory)* lub odczytać go za pomocą manipulatora *(Manipulator -> Tryb serwisowy -> Opcje -> Klucz integracji)*.

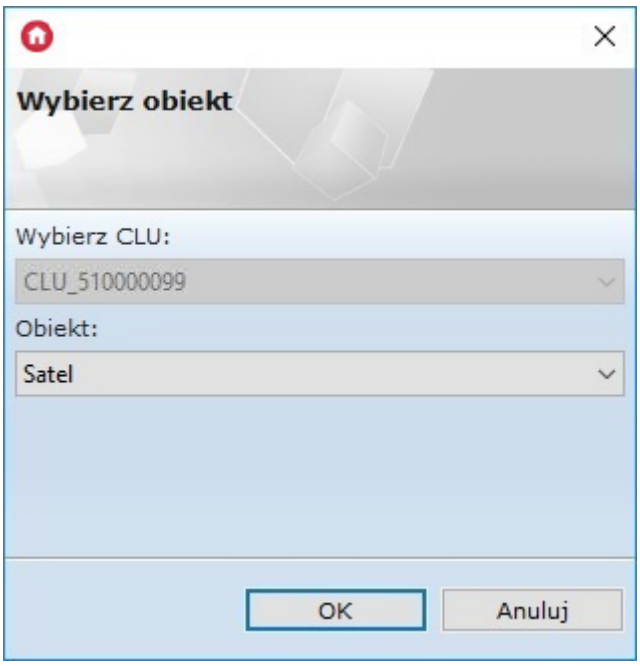


3.3. Obiekty wirtualne

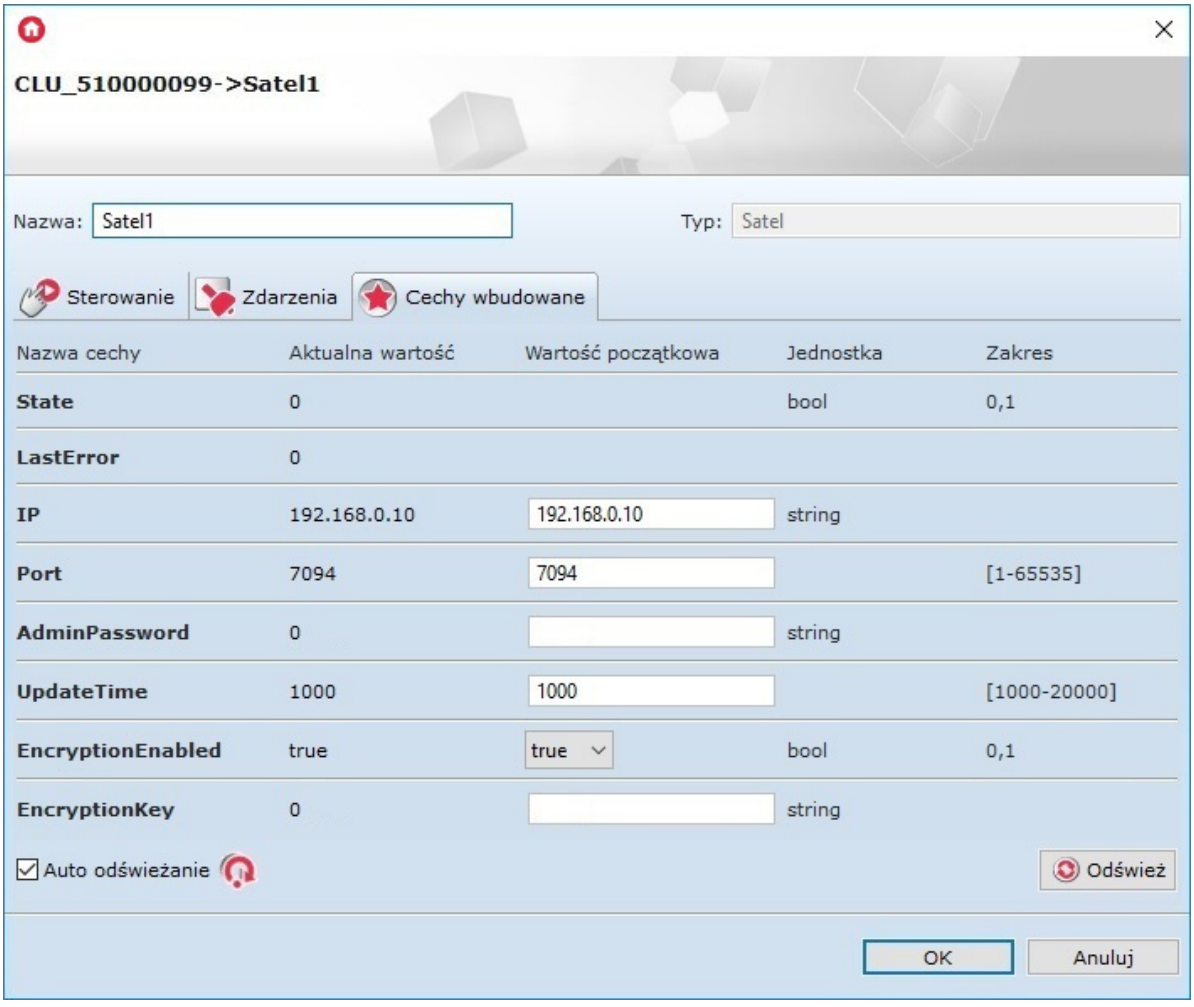
A. Satel

Chcąc przeprowadzić poprawną konfigurację modułu GATE Alarm należy kolejno:

- Utworzyć obiekt wirtualny *Satel*:

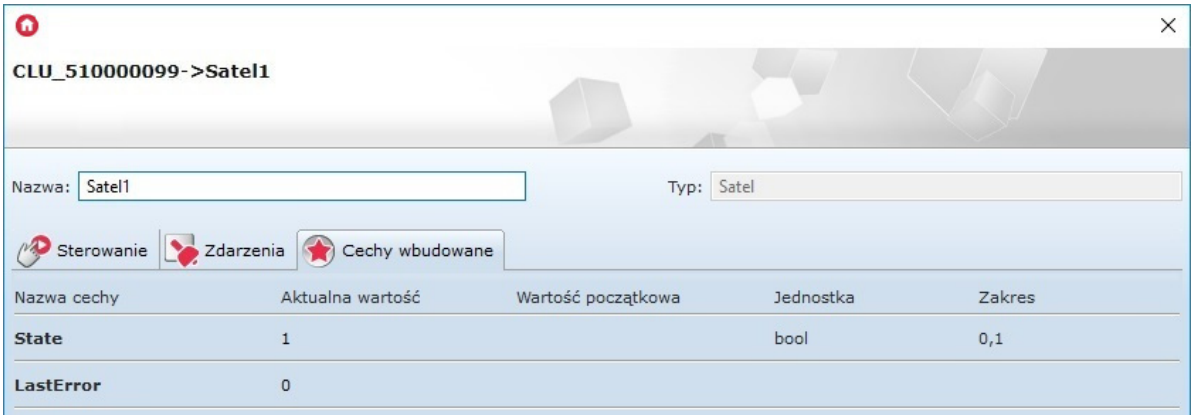


- Przejść do konfiguracji – zakładka *Cechy wbudowane* i wprowadzić:
  - **IP** – adres IP modułu ETHM (Satel);
  - **Port** – port integracji ETHM;
  - **AdminPassword** – hasło administratora;
  - **EncryptionEnabled** - załączenie kodowania – ustawić w sytuacji, gdy integracja na module ETHM ma zaznaczoną opcję *Kodowanie integracji*;
  - **Encryption Key** – klucz integracji (dla załączonego kodowania):



Informacje na temat tego, gdzie można znaleźć potrzebne informacje znajdują się w drugim podrozdziale - patrz pkt XIII.1.2..

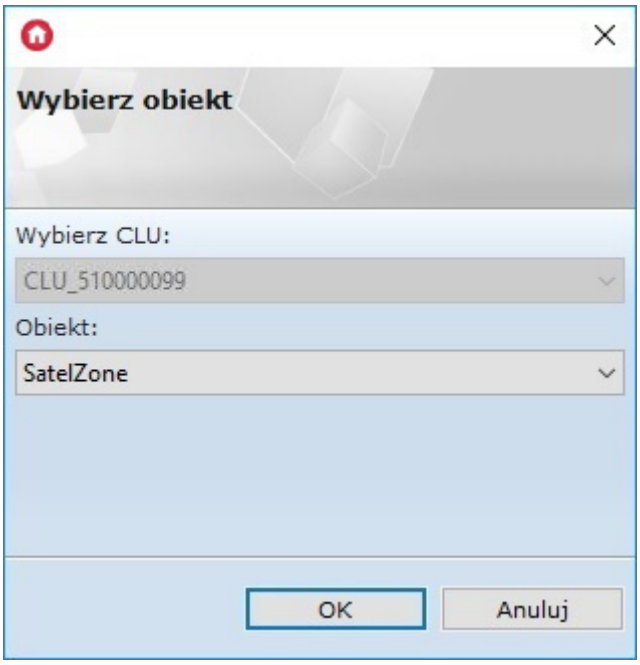
- Wysłać konfigurację oraz zweryfikować połączenie - zakładka *Cechy wbudowane*, cecha `State` (1 - poprawnie połączono z centralą, 0 - brak połączenia):



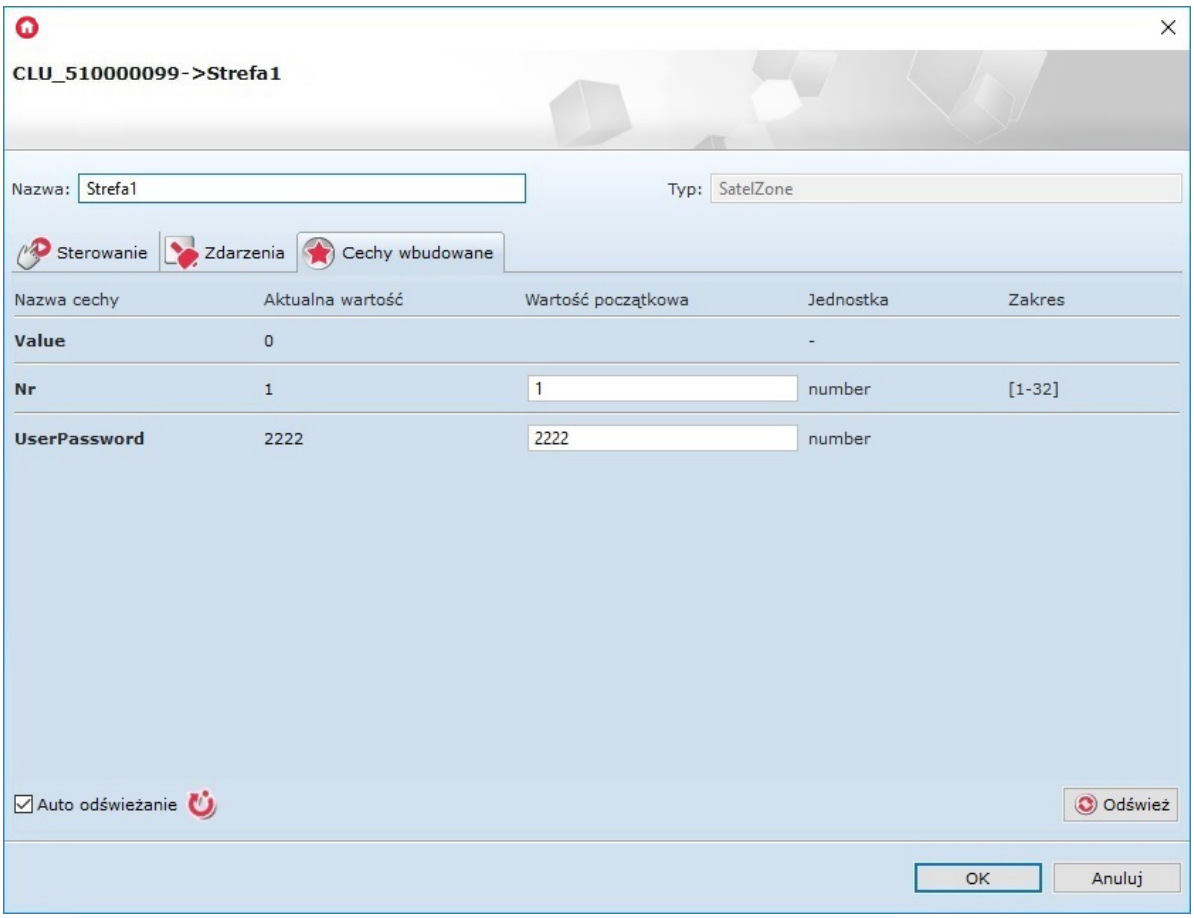
B. Strefa

Moduł GATE Alarm umożliwia dodanie obiektu wirtualnego *Strefa*:

- Utworzyć obiekt *SatelZone*:



- Zdefiniować Nr (numer wybranej strefy) oraz podać hasło użytkownika:

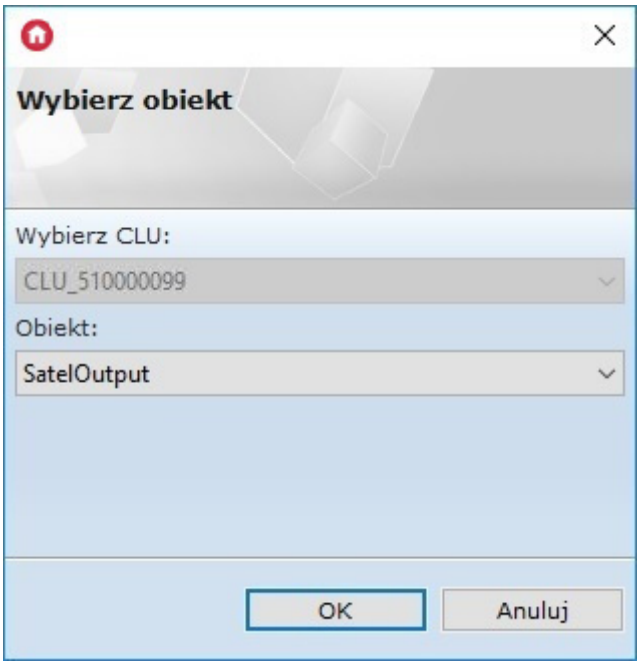


- Wysłać konfigurację oraz zweryfikować połączenie - zakładka *Cechy wbudowane*, cecha `Value` (-1 to brak połączenia z centralką, pozostałe oznaczają poprawne połączenie i zwracany jest stan strefy: 0 lub 1);
- Uzbroić/rozbroić strefę - metody `ArmZone` oraz `DisarmZone`.

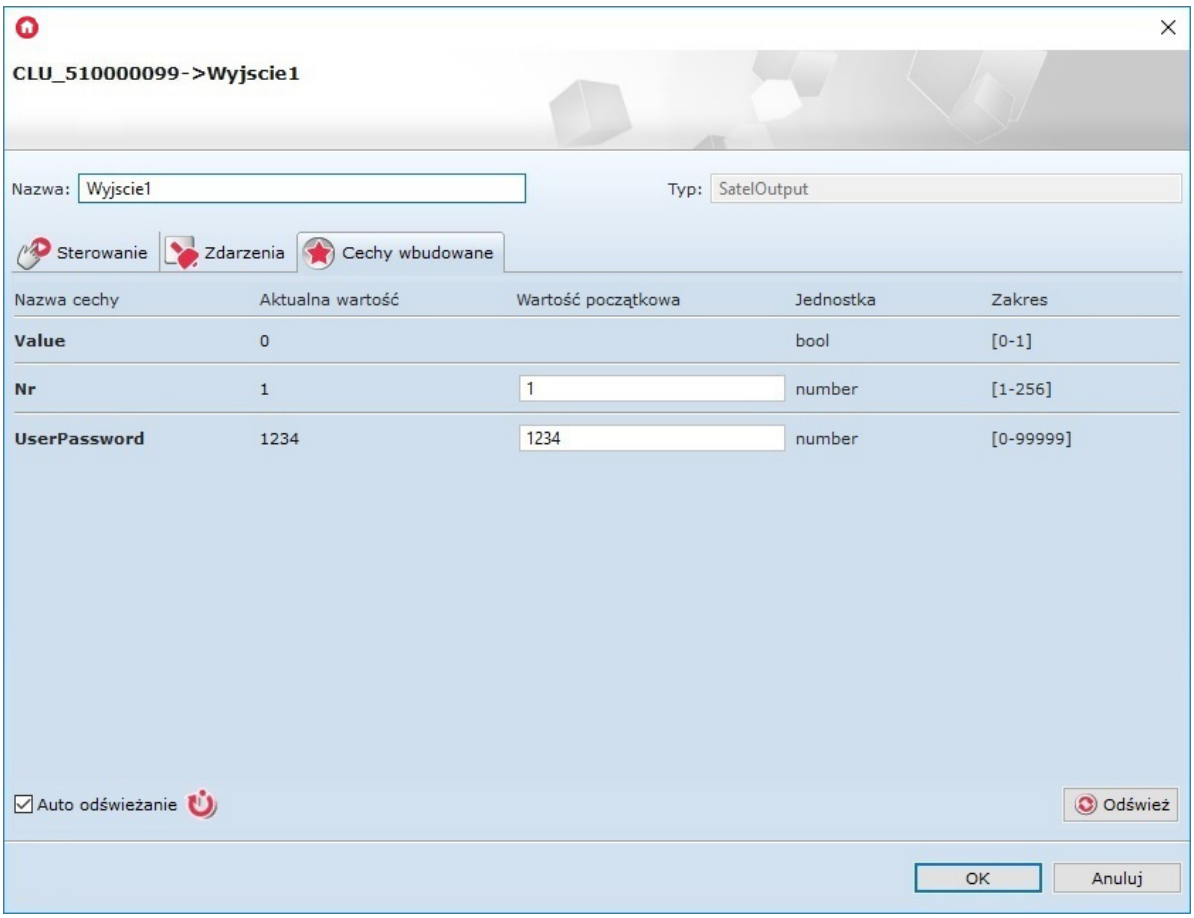
C. Wyjście

GATE Alarm umożliwia dodanie wirtualnego obiektu *Wyjście*:

- Utworzyć obiekt *SatelOutput*:



- Zdefiniować Nr (numer wybranego wyjścia na płytce Satel) oraz podać hasło użytkownika:

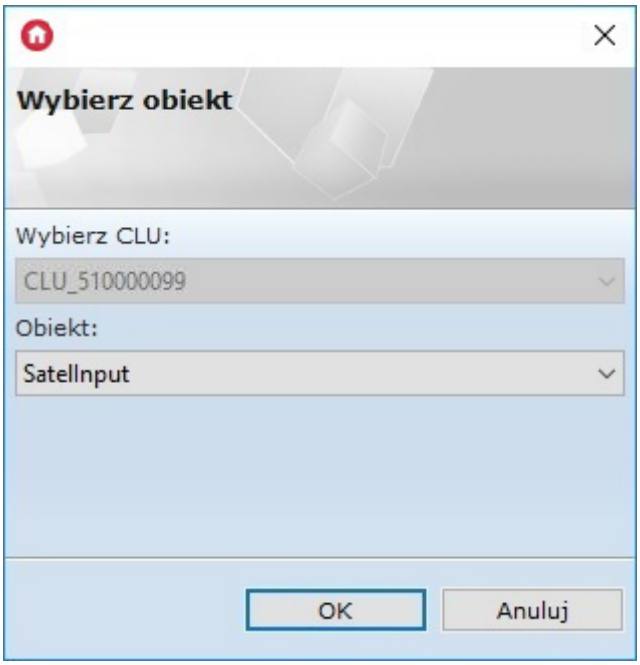


- Wysłać konfigurację oraz zweryfikować połączenie - zakładka *Cechy wbudowane*, cecha `Value` (-1 to brak połączenia z centralką, pozostałe oznaczają poprawne połączenie i zwracany jest stan strefy: 0 lub 1);
- Załączyć/wyłączyć wyjście - metody `SwitchOn` i `SwitchOff`.

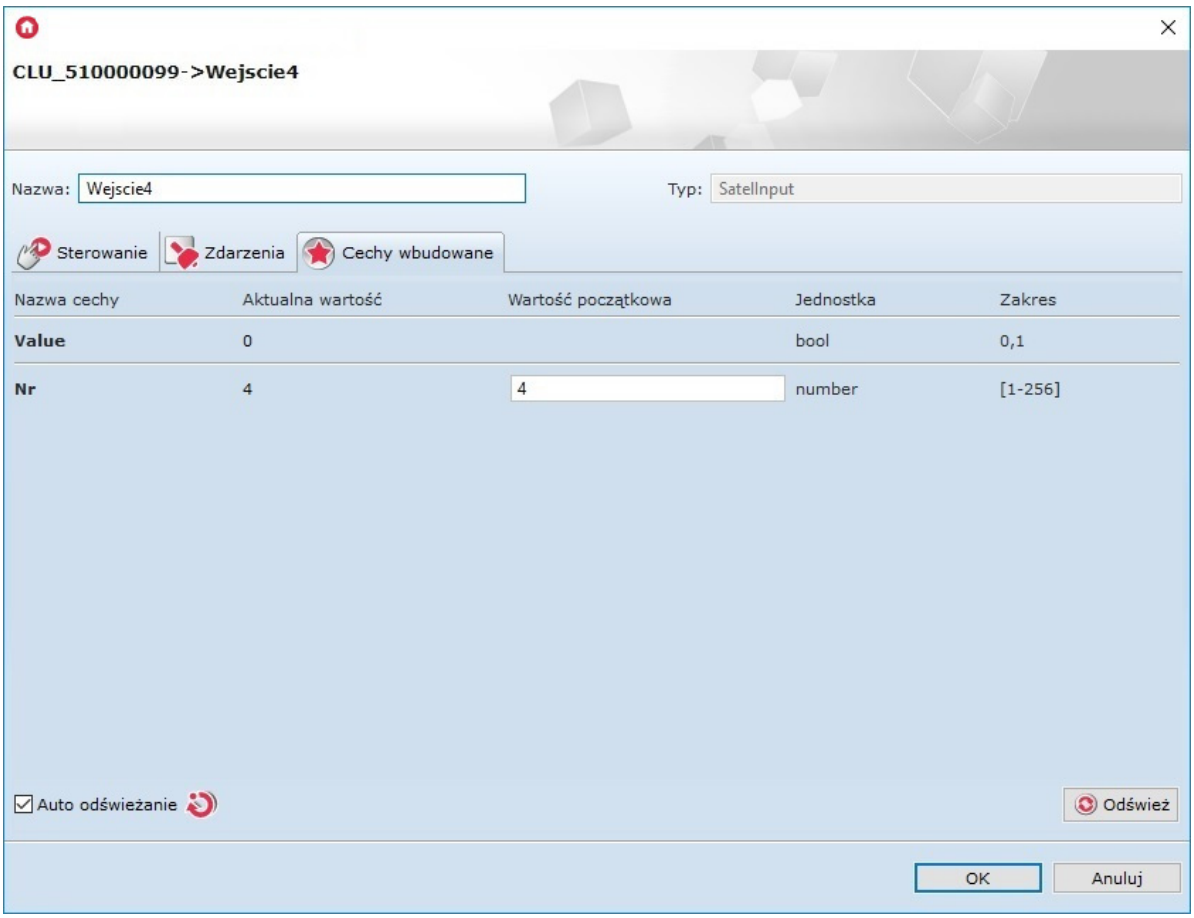
D. Wejście

GATE Alarm umożliwia dodanie wirtualnego obiektu *Wejście*:

- Utworzyć obiekt *SatelInput*:



- Zdefiniować Nr (numer wybranego wejścia na płytce Satel):



- Wysłać konfigurację oraz zweryfikować połączenie - zakładka *Cechy wbudowane*, cecha `Value` (-1 to brak połączenia z centralką, pozostałe oznaczają poprawne połączenie i zwracany jest stan strefy: 0 lub 1).

4. Integracja z centralą Jablotron

4.1. Informacje ogólne

Uwaga! Opisana funkcjonalność oraz integracja z wymienionymi centralami alarmowymi jest dostępna dla **GRENTON GATE ALARM, DIN, Eth (INT-221-E-01)** posiadający **firmware 1.1.20-2219 lub wyższy!**

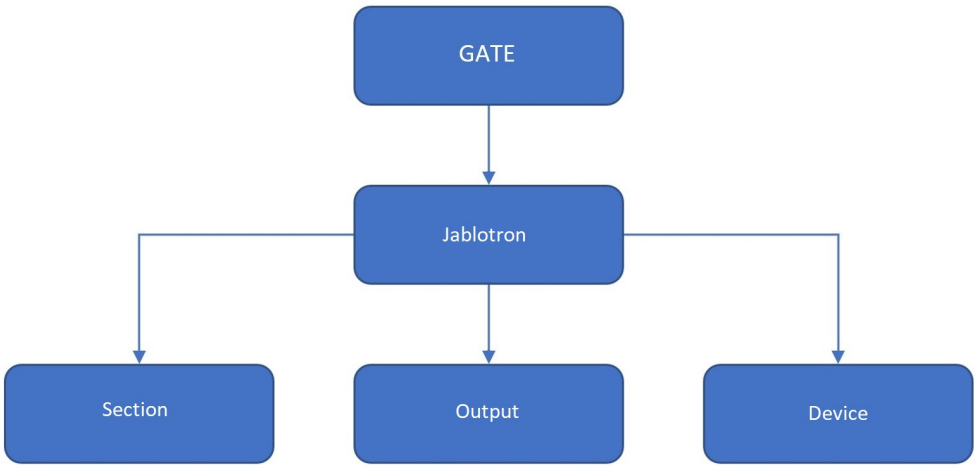
Uwaga! Integracja systemu Grenton dotyczy centrali alarmowych:

- **Jablotron JA-100K**,
- **Jablotron JA-101K**,
- **Jablotron JA-103K**,
- **Jablotron JA-107K**.

Integracja systemu Grenton z centralą alarmową Jablotron jest możliwa za pośrednictwem modułu JA-121T. Istnieje możliwość tworzenia obiektów wirtualnych typu: *JablotronSection, JablotronOutput, JablotronDevice.*

Uwaga! Dla tworzonych obiektów wirtualnych nie ma ograniczenia względem ilości obiektów - ograniczeniem jest pamięć urządzenia, na którą wpływa m.in. poziom rozbudowania logiki na module. Wyjątkiem jest obiekt **Jabloton** - można stworzyć tylko jeden.

Konstrukcja konfiguracji wygląda następująco:



- **Jablotron** – umożliwia przeprowadzenie konfiguracji pozwalającej na integrację systemu z centralą alarmową Grenton;
- **JablotronSection** – pozwala utworzyć strefę, do której dostęp będzie możliwy po wprowadzeniu hasła jednego z użytkowników lub hasła samego administratora;
- **JablotronDevice** – daje możliwość monitorowania stanu wybranego wejścia/urządzenia;
- **JablotronOutput** – pozwala monitorować oraz ustawiać stan wybranego wyjścia po podaniu hasła użytkownika lub administratora.

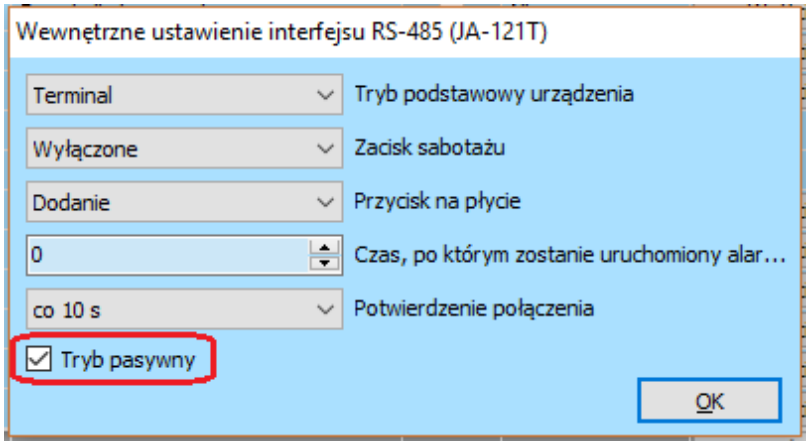
4.2. Konfiguracja dla systemu Jablotron

> **Uwaga!** Przed rozpoczęciem jakiejkolwiek pracy z modułem GATE Alarm wymagana jest aktualizacja bazy interfejsów!

Do integracji pomiędzy modułem Gate a centralką alarmową Jablotron, wykorzystywany jest modułu **JA-121T**. Moduł JA-121T musi być podłączony do systemu centralki Jablotron i dodany do listy systemu. Informacje odnośnie dodawania / użytkowania poszczególnych modułów Jablotron znajdują się w dokumentacji producenta.

Komunikacja pomiędzy Gate a modułem JA-121T odbywa się za pośrednictwem interfejsu RS485 – połączenie pomiędzy zaciskami śrubowymi A (D+), B (D-) na obydwu modułach.

> **Uwaga!** Moduł JA-121T musi mieć załączony w ustawieniach Tryb pasywny.



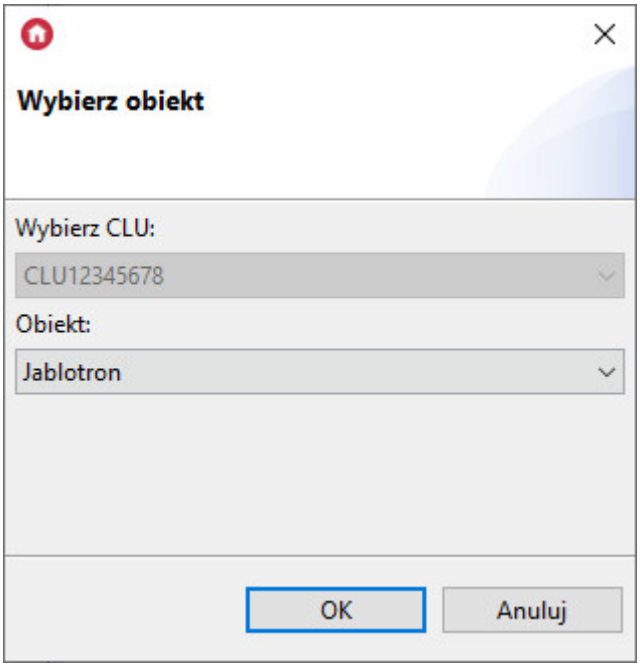
> **Uwaga!** Informacje o pozostałych ustawieniach i ich zastosowaniu znajdują się w dokumentacji modułu JA-121T na stronie producenta.

Po wykonaniu podłączenia modułów oraz prawidłowego skonfigurowania urządzenia JA-121T, można przystąpić do tworzenia i konfiguracji obiektów wirtualnych modułu GATE Alarm.

4.3. Obiekty wirtualne

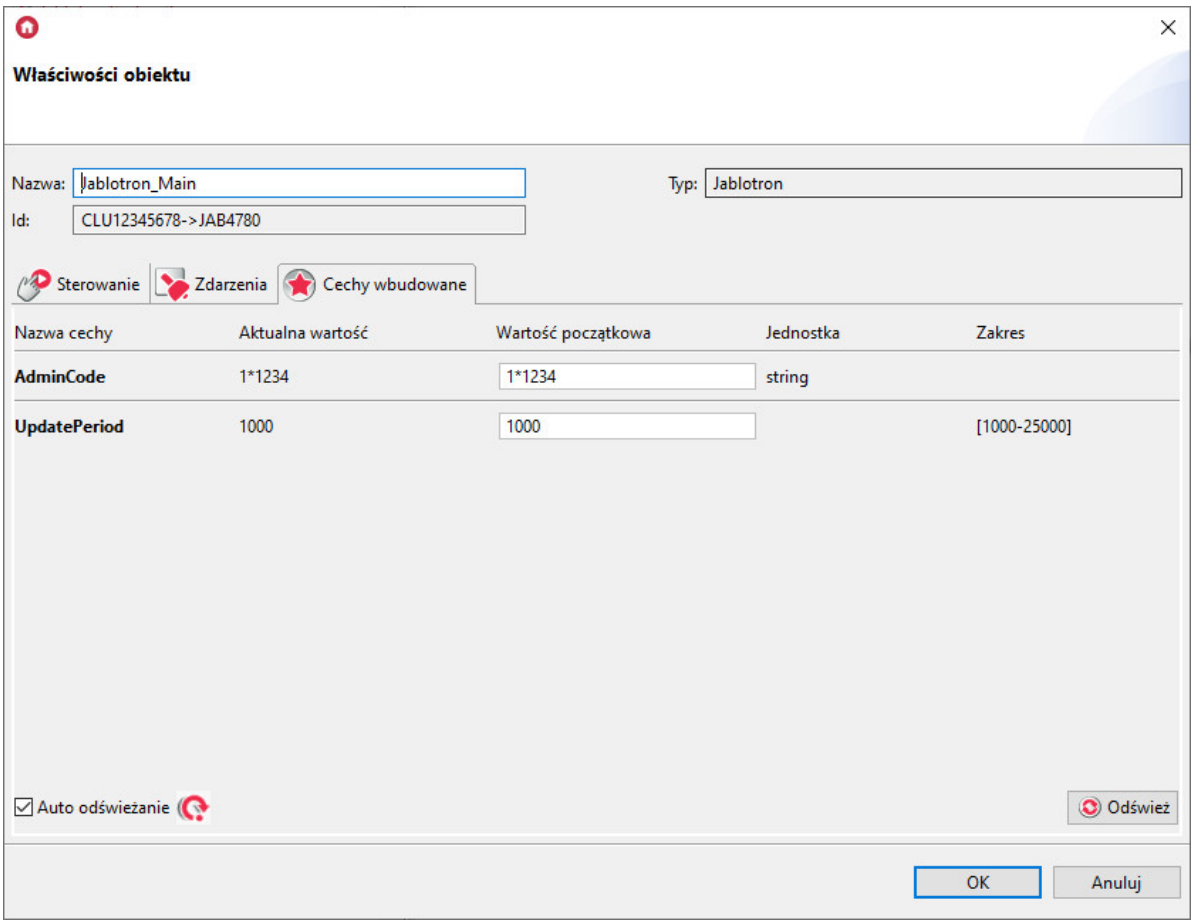
A. Jablotron

W celu przeprowadzenia poprawnej konfiguracji modułu GATE Alarm należy utworzyć obiekt wirtualny **Jablotron**:



Następnie, przejść do konfiguracji - zakładka Cechy wbudowane i wprowadzić

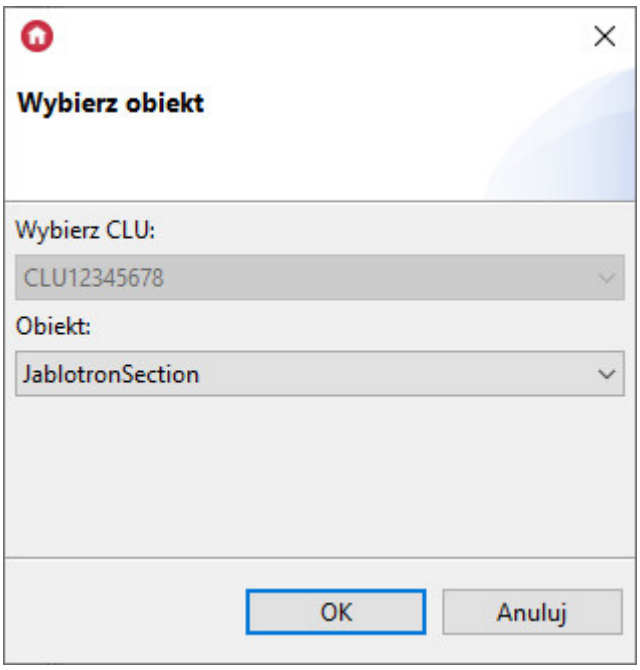
- **AdminCode** - kod dostępu administratora;
- **UpdatePeriod** - Okres aktualizacji stanu centralki;



B. Sekcja/strefa

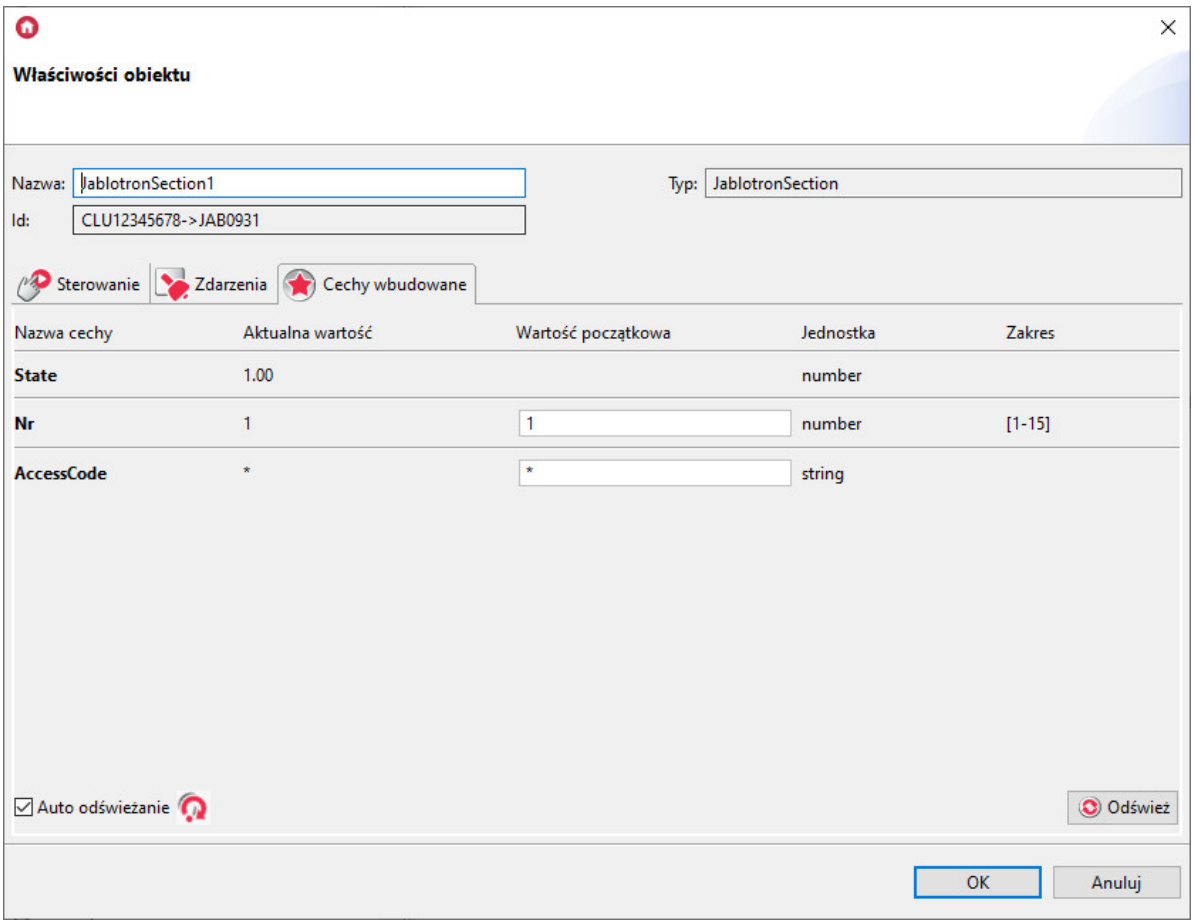
GATE Alarm umożliwia dodanie wirtualnego obiektu **JablotronSection**:

Tworzenie obiektu *JablotronSection*:



Następnie, przejść do konfiguracji - zakładka Cechy wbudowane i wprowadzić:

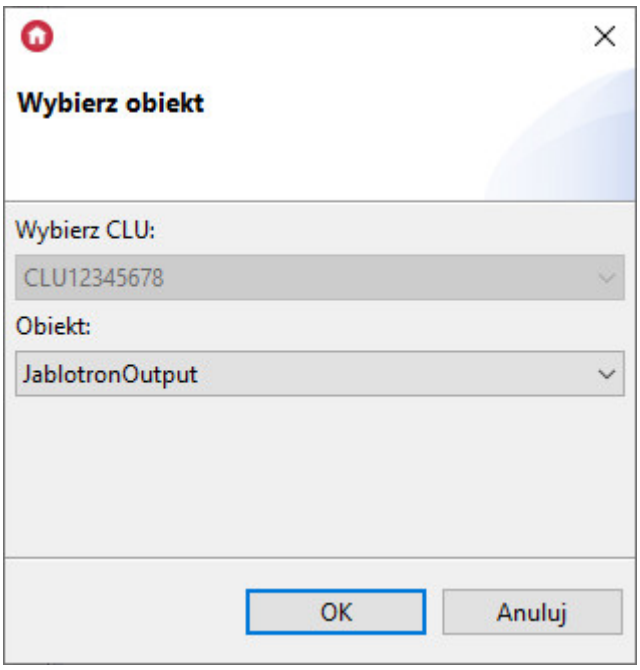
- **Nr** – parametr definiujący, do której sekcji odnosi się obiekt;
- **UpdatePeriod** – Kod dostępu, dla '*' zastosuje kod administratora;



C. Wyjście

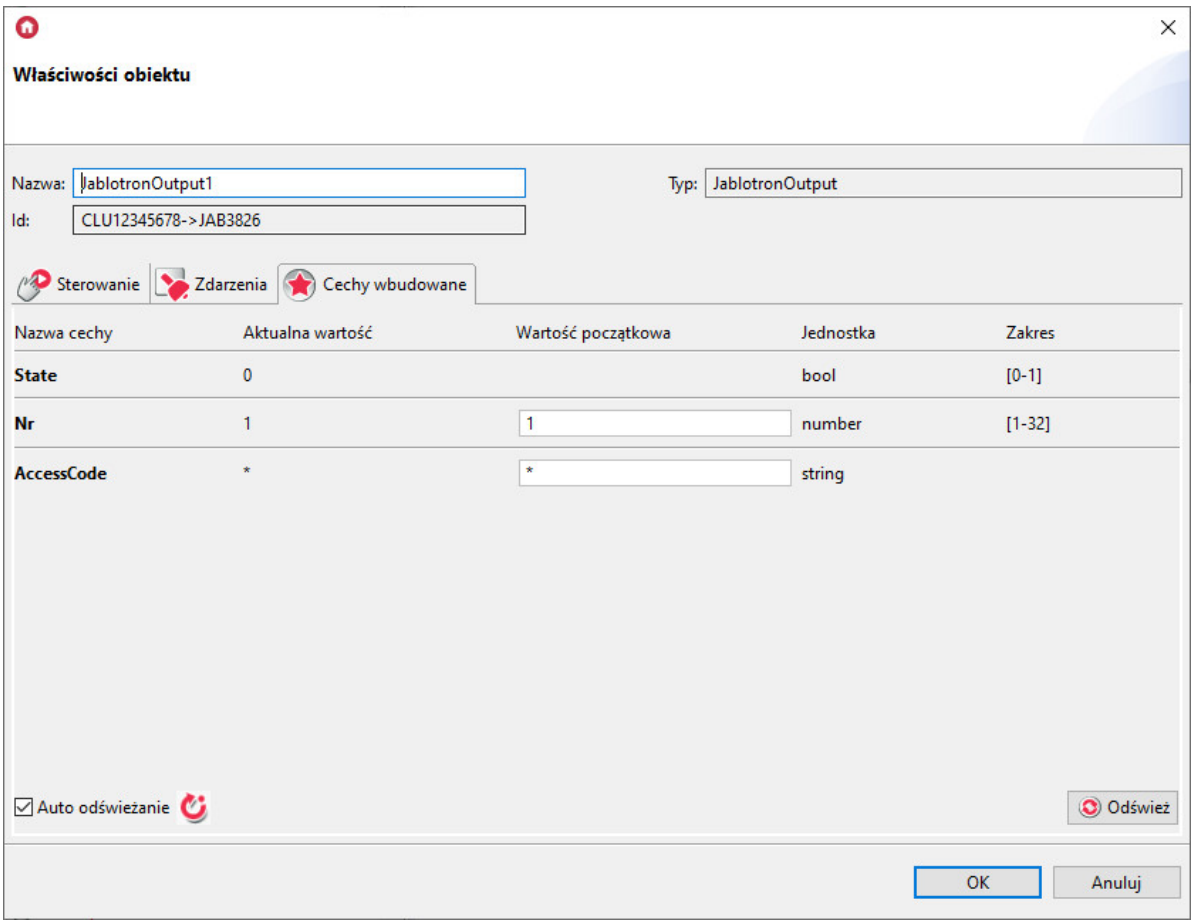
GATE Alarm umożliwia dodanie wirtualnego obiektu **JablotronOutput**

Tworzenie obiektu *JablotronOutput*:



Następnie, przejść do konfiguracji - zakładka Cechy wbudowane i wprowadzić:

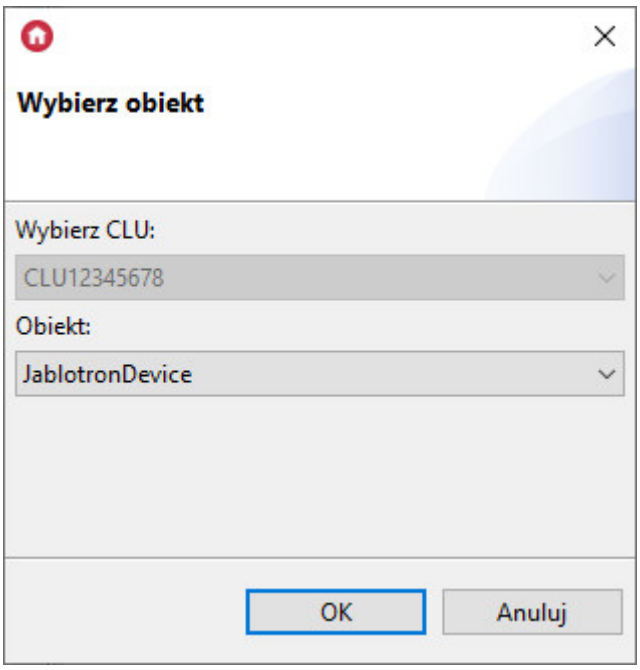
- **Nr** – parametr definiujący, do którego wyjścia odnosi się obiekt;
- **AccessCode** – Kod dostępu, dla '*' zastosuje kod administratora;



D. Wejście/Urządzenie

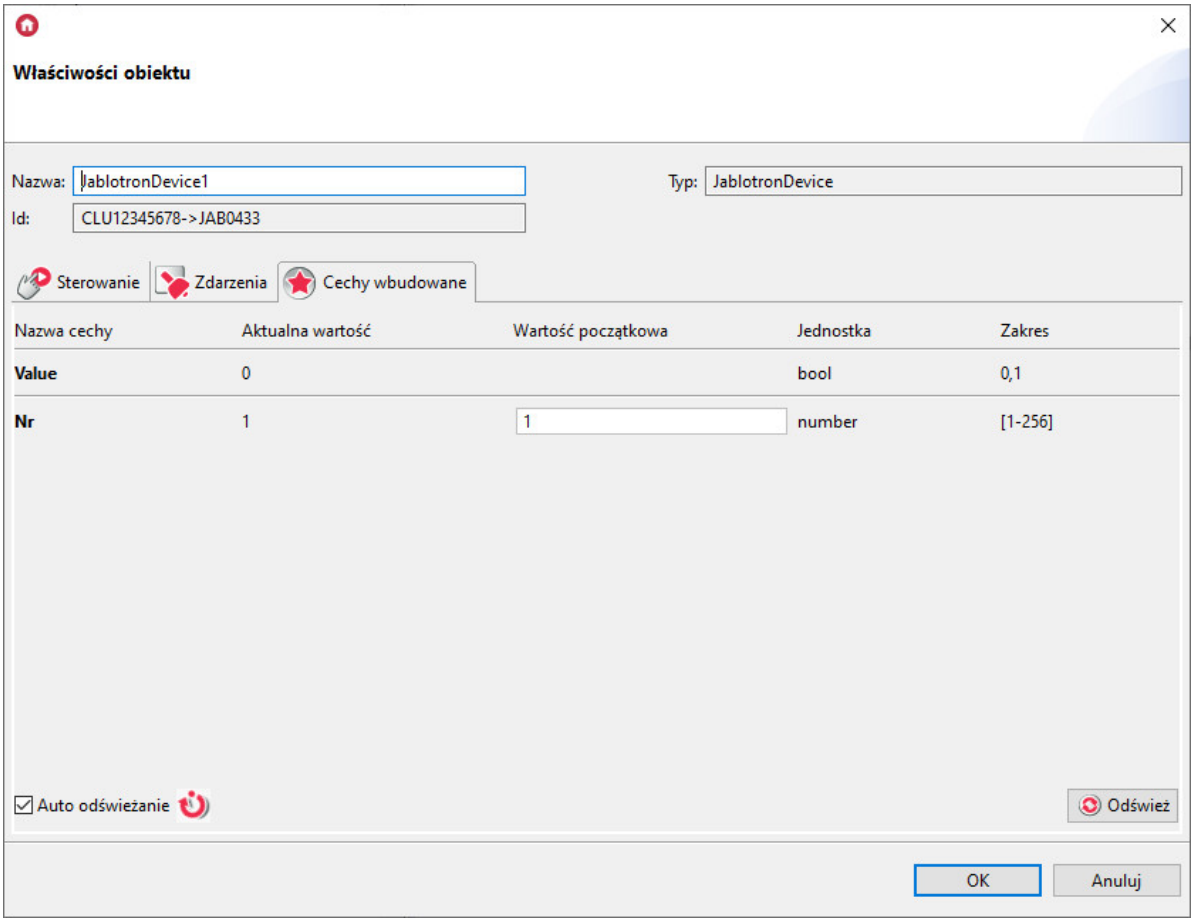
GATE Alarm umożliwia dodanie wirtualnego obiektu **JablotronDevice**

Tworzenie obiektu *JablotronDevice*:



Następnie, przejść do konfiguracji - zakładka Cechy wbudowane i wprowadzić:

- **Nr** – parametr definiujący, do którego wejścia odnosi się obiekt;



5. Obiekt wirtualny - Timer

Timery są wirtualnymi obiektami tworzonymi w ramach danego modułu GATE. Timery mogą być wykorzystywane wszędzie tam, gdzie potrzebne jest wywołanie metody po określonym czasie lub też jej cykliczne wywoływanie.

Timer może pracować w dwóch trybach:

- `Countdown` Po wystartowaniu, odlicza ustalony czas. Po zakończeniu odliczania uruchamiana jest metoda powiązana ze zdarzeniem `OnTimer`, a timer zatrzymuje się i nie odlicza, aż do następnego uruchomienia metodą `Start`.
- `Interval` Timer cykliczny - po starcie zaczyna odliczać ustawiony czas. Po jego upłynięciu timer wywołuje metodę powiązaną ze zdarzeniem `OnTimer`, a sam timer ponownie zaczyna odliczać zadany czas. Sytuacja powtarza się, aż do momentu zatrzymania metodą `Stop`.

6. Przywracanie ustawień fabrycznych - *Hard Reset*

Uruchomienie funkcji *Hard Reset* na module GATE Alarm powoduje:

- Usunięcie zapisanej konfiguracji;
- Sformatowanie partycji pamięci flash;
- Usunięcie wszystkich utworzonych obiektów LUA;
- Utratę komunikacji pomiędzy OM / HM a modułem Gate.

W celu przywrócenia ustawień fabrycznych funkcją *Hard Reset* należy wykonać następujące czynności (zgodnie z podaną kolejnością):

- Odłączyć zasilanie od modułu Gate
- Nacisnąć i przytrzymać przycisk *Reset* na module (przycisk znajduje się pod dolną zaślepką modułu)
- Podłączyć zasilanie do modułu Gate
- Trzymać wciśnięty przycisk *Reset* przez co najmniej 10 sekund - podczas resetu dioda zielona będzie świecić światłem ciągłym. Prawidłowe wykonanie resetu zostanie potwierdzone 3-krotnym mrugnięciem diody zielonej
- Po upływie 10 sekund zwolnić przycisk *Reset*
- Odczekać około 60 sekund aż do momentu, gdy na module diody - zielona oraz czerwona - będą mrugać naprzemiennie (tryb *Emergency*).

Po wykonaniu procedury na module zostanie wyczyszczona konfiguracja, natomiast sam moduł przestanie być widoczny (brak odpowiedzi na *Keep-Alive*) w projekcie z poziomu Object Managera. Aby ponownie przywrócić moduł, należy wykonać CLU Discovery a następnie wysłać konfigurację.

7. Parametry konfiguracyjne

> **Uwaga!** Opisana funkcjonalność oraz integracja z wymienionymi centralami alarmowymi jest dostępna dla **GRENTON GATE ALARM, DIN, Eth (INT-221-E-01)** posiadający **firmware 1.1.20-2219 lub wyższy!**

A. GATE

CECHY

Nazwa	Opis
`Uptime`	Czas pracy urządzenia od ostatniego resetu (w sekundach)
`ClientReportInterval`	Okres raportowania o zmianach cech
`Date`	Aktualna data
`Time`	Aktualny czas (hh:mm:ss)
`LocalTime`	Aktualny lokalny znacznik czasu
`TimeZone`	Strefa czasowa
`UnixTime`	Aktualny uniksowy znacznik czasu
`FirmwareVersion`	Wersja oprogramowania gate
`UseCloud`	Określa czy GATE łączy się do chmury
`CloudConnection`	Określa status połączenia GATE z chmurą
`NTPTimeout`	Czas oczekiwania na odpowiedź z serwera NTP
`UseNTP`	Określa czy GATE używa NTP
`PrimaryDNS`	Preferowany serwer DNS
`SecondaryDNS`	Alternatywny serwer DNS

METODY

Nazwa	Opis
`SetDateTime`	Ustawia datę i czas
`StartConsole`	Uruchamia konsolę Lua
`StartConsoleOnReboot`	Uruchamia konsolę Lua przy ponownym uruchomieniu
`SetClientReportInterval`	Ustawia okres raportowania o zmianach cech
`SetPrimaryDNS`	Ustawia cechę PrimaryDNS
`SetSecondaryDNS`	Ustawia cechę SecondaryDNS

ZDARZENIA

Nazwa	Opis
`OnInit`	Zdarzenie wywoływane w momencie inicjalizacji urządzenia

B. Satel

CECHY

Nazwa	Opis
`State`	Stan centralki (0 – brak połączenia z centralką, 1 – połączono z centralką)
`LastError`	Ostatni kod błędu modułu ETHM (0 – ok, 1 – nieprawidłowe hasło)
`IP`	Adres IP modułu ETHM (Satel)
`Port`	Port modułu ETHM (Satel)
`AdminPassword`	Hasło administratora Satel
`EncryptionEnabled`	Stan szyfrowania (*true* - załączone, *false* - wyłączone)
`EncryptionKey`	Klucz szyfrowania Satel
`Value`	Zwraca aktualny stan (1 – dla strefy uzbrojonej, wejścia naruszonego, wyjścia załączonego; 0 – dla strefy rozbrojonej, wejścia nienaruszonego, wyjścia wyłączonego; -1 – brak informacji na temat stanu z powodu braku połączenia)
`Nr`	Parametr definiujący strefę, wejście lub wyjście, do których odnosi się obiekt
`UserPassword`	Hasło użytkownika (dla „_” zastosuje hasło administratora)

METODY

Nazwa	Opis
`SetIP`	Ustawia adres IP modułu ETHM (Satel)
`SetPort`	Ustawia port modułu ETHM (Satel)
`SetAdminPassword`	Ustawia hasło administratora
`SetEncryptionEnabled`	Włącza/wyłącza szyfrowanie
`SetEncryptionKey`	Ustawia klucz szyfrowania Satel
`ArmZone`	Uzbraja strefę
`DisarmZone`	Rozbraja strefę
`SetNr`	Ustawia parametr definiujący, do której strefy, wejścia lub wyjścia odnosi się obiekt
`SetUserPassword`	Ustawia hasło użytkownika (dla _ zastosuje hasło administratora)
`SwitchOn`	Załącza wyjście
`SwitchOff`	Wyłącza wyjście

ZDARZENIA

Nazwa	Opis
`OnConnected`	Zdarzenie wywoływane po nawiązaniu połączenia z centralą
`OnDisconnected`	Zdarzenie wywołane po utraceniu połączenia z centralką
`OnError`	Zdarzenie wywołane po wystąpieniu błędu centralki (`LastError`)
`OnChange`	Zdarzenie wywoływane przy zmianie stanu (niezależnie od wartości)
`OnSwitchOn`	Zdarzenie wywołane przy załączeniu wyjścia lub naruszeniu wejścia
`OnSwitchOff`	Zdarzenie wywołane przy wyłączeniu wyjścia lub ustawieniu stanu normalnego na wejściu
`OnArm`	Zdarzenie wywołane przy uzbrojeniu strefy
`OnDisarm`	Zdarzenie wywołane przy rozbrojeniu strefy

C. Jablotron

CECHY

Nazwa	Opis
`AdminCode`	Kod dostępu Administratora
`State`	*JablotronSection:* 1 - READY - normalny tryb pracy, 2 - ARMED_PART - częściowe uzbrojenie sekcji, 3 - ARMED - sekcja uzbrojona, 4 - SERVICE - załączony tryb serwisowy, 5 - BLOCKED - sekcja zablokowana, 6 - OFF - sekcja wyłączona *JablotronOutput*: 0 - Wyjście załączone 1 - Wyjście wyłączone
`Nr`	Parametr definiujący, do którego wejścia/wyjścia/sekcji odnosi się obiekt
`AccessCode`	Kod dostępu, dla '*' zastosuje kod administratora
`Value`	Zwraca aktualny stan: 1 – dla strefy uzbrojonej, wejścia naruszonego, wyjścia załączonego; 0 – dla strefy rozbrojonej, wejścia nienaruszonego, wyjścia wyłączonego; -1 – brak informacji na temat stanu z powodu braku połączenia

METODY

Nazwa	Opis
`SetAccessCode`	Ustawia kod dostępu, dla '*' zastosuje kod administratora
`Arm`	Uzbraja strefę/sekcję
`ArmPartially`	Częściowo uzbraja sekcję (Jablotron)
`Disarm`	Rozbraja strefę/sekcję
`SetNr`	Ustawia parametr definiujący, do której strefy, wejścia lub wyjścia odnosi się obiekt
`SwitchOn`	Załącza wyjście
`SwitchOff`	Wyłącza wyjście

ZDARZENIA

Nazwa	Opis
`OnStateChange`	Zdarzenie wywoływane przy zmianie stanu (niezależnie od wartości)
`OnArm`	Zdarzenie wywoływane przy uzbrojeniu sekcji
`OnDisarm`	Zdarzenie wywoływane przy rozbrojeniu sekcji
`OnChange`	Zdarzenie wywoływane przy zmianie stanu (niezależnie od wartości)
`OnSwitchOn`	Zdarzenie wywołane przy załączeniu wyjścia lub naruszeniu wejścia
`OnSwitchOff`	Zdarzenie wywołane przy wyłączeniu wyjścia lub ustawieniu stanu normalnego na wejściu
`OnArm`	Zdarzenie wywołane przy uzbrojeniu strefy
`OnDisarm`	Zdarzenie wywołane przy rozbrojeniu strefy

D. Timer

CECHY

Nazwa	Opis
`Time`	Zliczany czas (w ms)
`Mode`	Tryb pracy timera: 0 – zliczenie w dół (countdown), 1 – cykliczny (interval)
`State`	Aktualny stan pracy timera: 0 – zatrzymany (stopped), 1 – liczy (counting)

METODY

Nazwa	Opis
`SetTime`	Ustawia czas timera (w ms)
`SetMode`	Ustawia tryb pracy: 0 – zliczenie w dół (countdown), 1 – cykliczny (interval)
`Start`	Uruchamia timer
`Stop`	Zatrzymuje timer

ZDARZENIA

Nazwa	Opis
`OnTimer`	Zdarzenie wywoływane przy zliczeniu timera
`OnStart`	Zdarzenie wywoływane przy uruchomieniu timera
`OnStop`	Zdarzenie wywoływane przy zatrzymaniu timera

XIV. Moduł GATE MODBUS

> **Uwaga!** Opisana funkcjonalność oraz integracja jest dostępna dla **GRENTON GATE MODBUS MASTER, DIN, Eth (INT-201-E-01)** posiadający **firmware 1.1.0-2034C lub wyższy!**

1. Informacje ogólne

Moduł GATE MODBUS umożliwia integrację systemu Grenton ze wszystkimi urządzeniami wspierającymi standard MODBUS RTU.

> **Uwaga!** Dla tworzonych obiektów wirtualnych nie ma ograniczenia względem ilości obiektów - ograniczeniem jest pamięć urządzenia, na którą wpływa m.in. poziom rozbudowania logiki na module.

Przed rozpoczęciem konfiguracji należy uzyskać informacje dotyczące wybranego urządzenia Slave wspierającego standard MODBUS RTU - konieczna będzie znajomość m.in.: adresu urządzenia, typu oraz adresów rejestrów, a także prędkości transmisji.

2. Konfiguracja modułu

> **Uwaga!** Przed rozpoczęciem jakiekolwiek pracy z modułem GATE Modbus wymagana jest aktualizacja bazy interfejsów!

2.1. Ustawianie czasu za pomocą serwera NTP

Moduł GATE Modbus umożliwia ustawianie czasu za pomocą serwera NTP wraz z uwzględnieniem strefy czasowej a także zmianą czasu (zimowy/letni). Czas pobierany jest automatycznie z serwera NTP (*pool.ntp.org*).

Do konfiguracji służą trzy cechy:

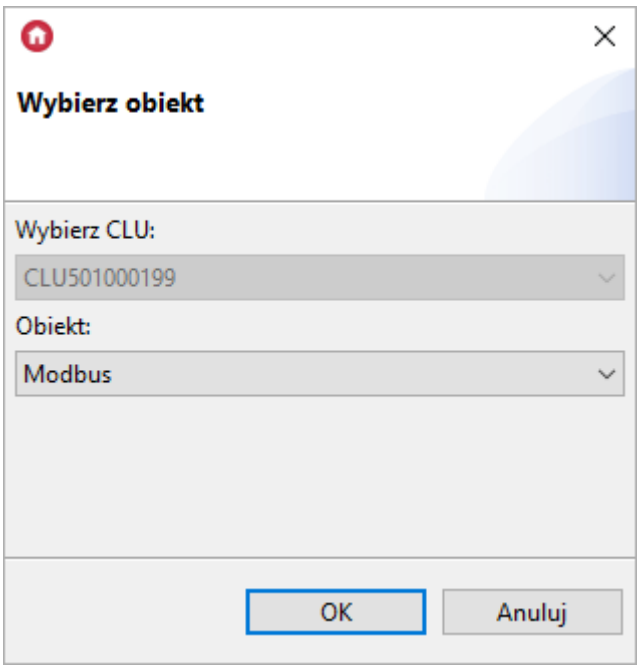
- `UseNTP` – określa czy GATE używa NTP,
- `NTPTimeout` - czas oczekiwania na odpowiedź z serwera NTP,
- `TimeZone` – ustawianie strefy czasowej GATE – dostępne są 22 strefy.

> **Uwaga!** Pobieranie czasu z serwera NTP wymaga, aby GATE znajdowało się w sieci, która posiada połączenie z internetem.

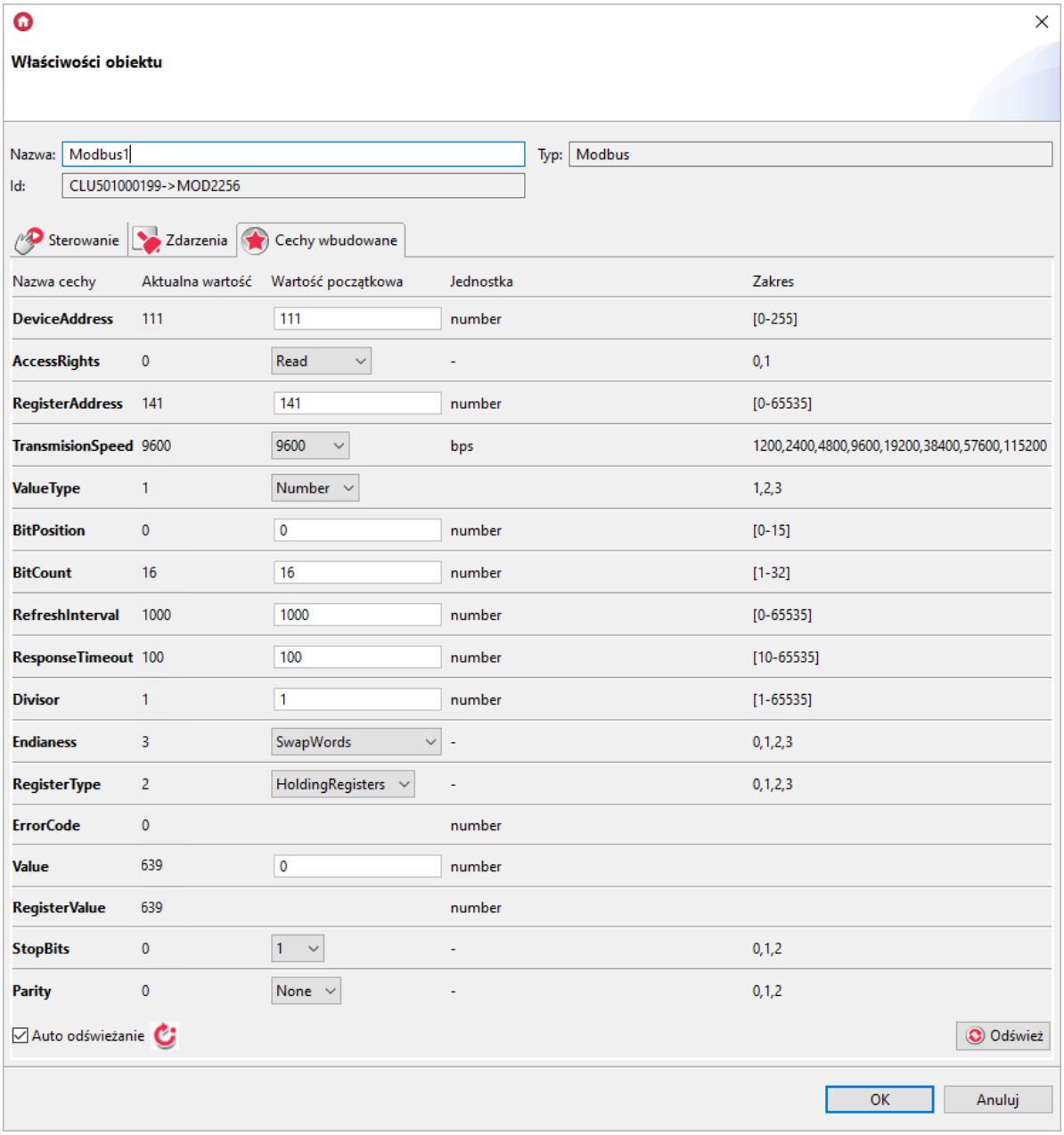
2.2. Konfiguracja obiektu wirtualnego Modbus

Chcąc przeprowadzić poprawną konfigurację modułu GATE Modbus należy kolejno:

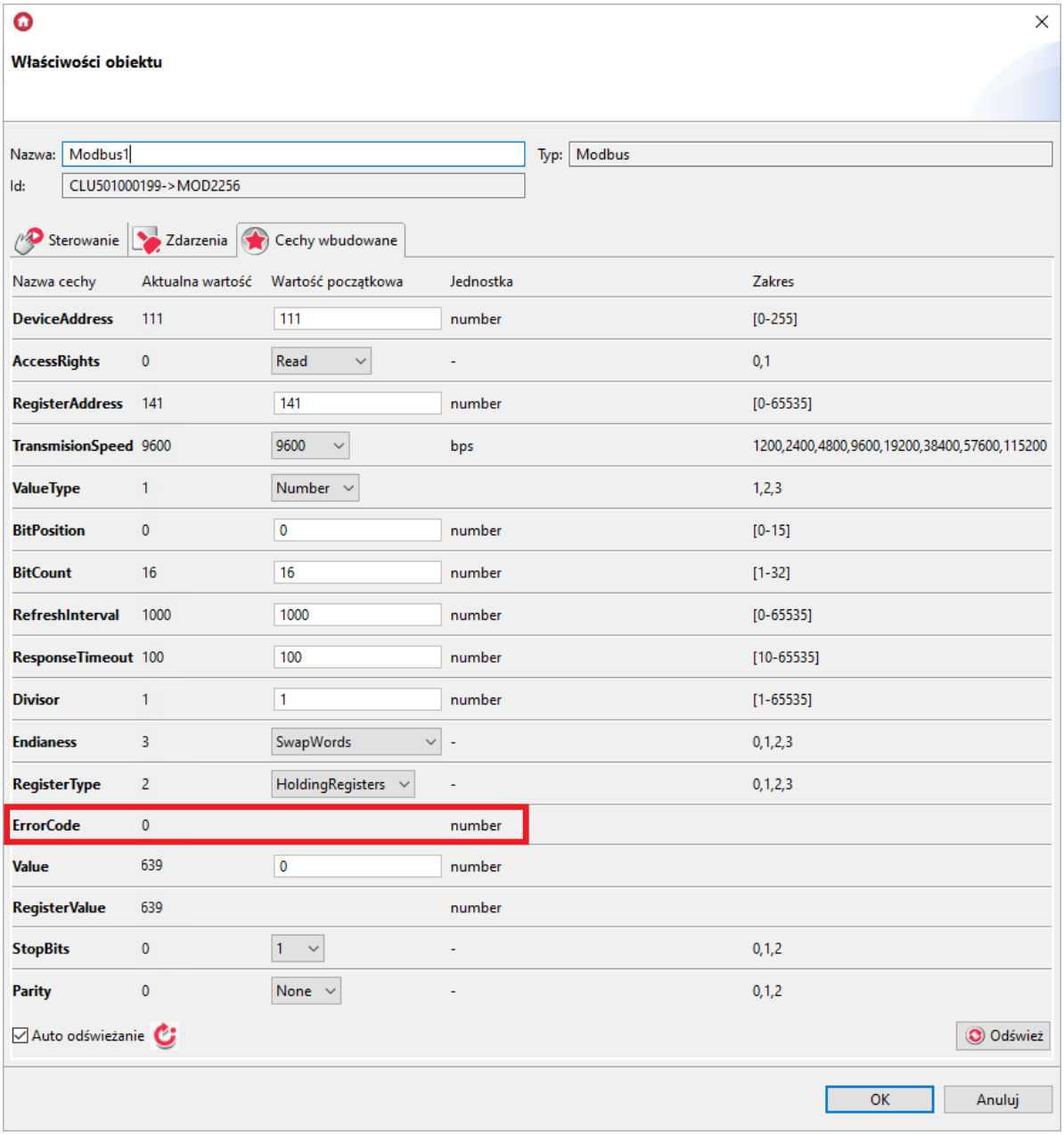
- Utworzyć obiekt wirtualny *Modbus* i nadać mu nazwę:



- Przejść do zakładki *Cechy wbudowane* i wprowadzić:
  - **DeviceAddress** – adres urządzenia Slave;
  - **AccessRights** - tryb pracy (*Read* - odczyt wartości z rejestru; *ReadWrite* - umożliwia zapis wartości do ustawionego rejestru);
  - **RegisterAddress** – adres obsługiwanego rejestru;
  - **TransmissionSpeed** – prędkość transmisji;
  - **RefreshInterval** - okres odpytywania rejestru urządzenia Slave przez GATE Modbus;
  - **ResponseTimeout** - czas urządzenia Slave na odpowiedź (po jego przekroczeniu zwracany jest `ErrorCode`=**-2**);
  - **Divisor** - dzielnik (dla `ValueType`=*Number/Float*);
  - parametry odpowiednie dla wybranego typu rejestru urządzenia Slave - patrz pkt.3.1.
  - **StopBits** - liczba bitów stopu urządzenia Slave: 0 - 1 bit stopu 1 - 1.5 bitu stopu 2 - 2 bity stopu
  - **Parity** - parametr ustawia sprawdzanie parzystości: 0 - None 1 - Odd 2 - Even



- Wysłać konfigurację oraz zweryfikować połączenie - zakładka *Cechy wbudowane*, cecha `ErrorCode` =0 (poprawny odczyt/zapis):

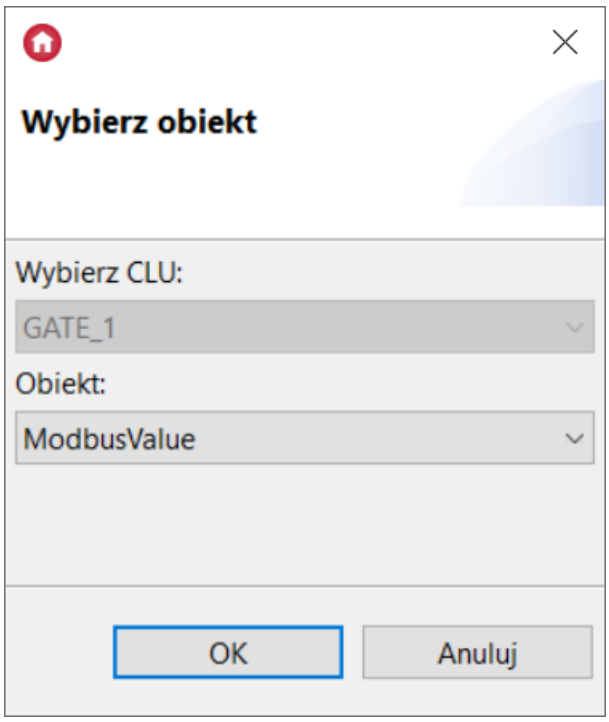


2.3. Konfiguracja obiektu wirtualnego ModbusValue

> **Uwaga!** Opisany obiekt wirtualny jest dostępny dla **GRENTON GATE MODBUS MASTER, DIN, Eth (INT-201-E-01)** posiadający **firmware 1.1.10-2140 lub wyższy!**

Aby użyć obiektu wirtualnego ModbusValue należy:

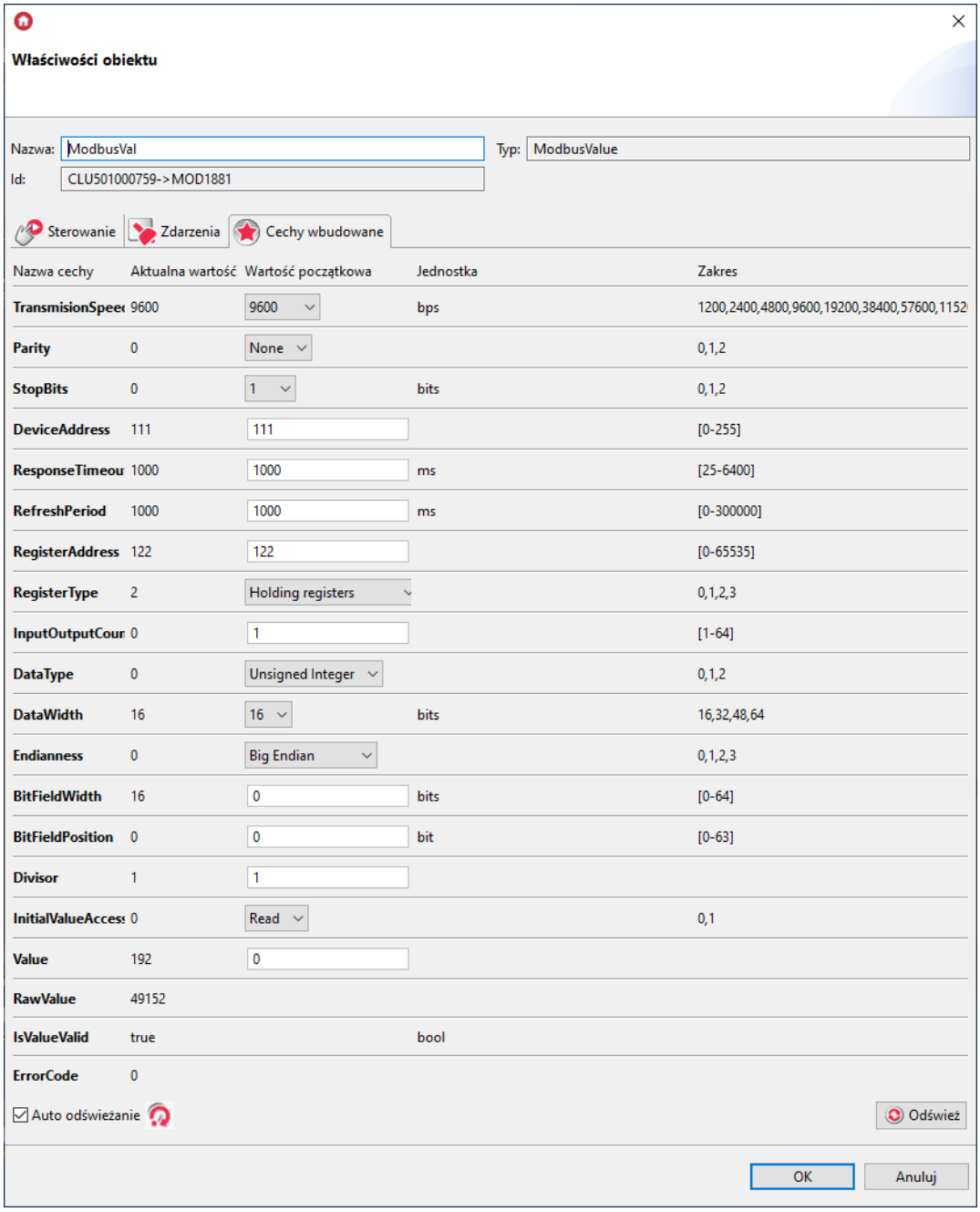
- Utworzyć obiekt wirtualny *ModbusValue* i nadać mu nazwę:



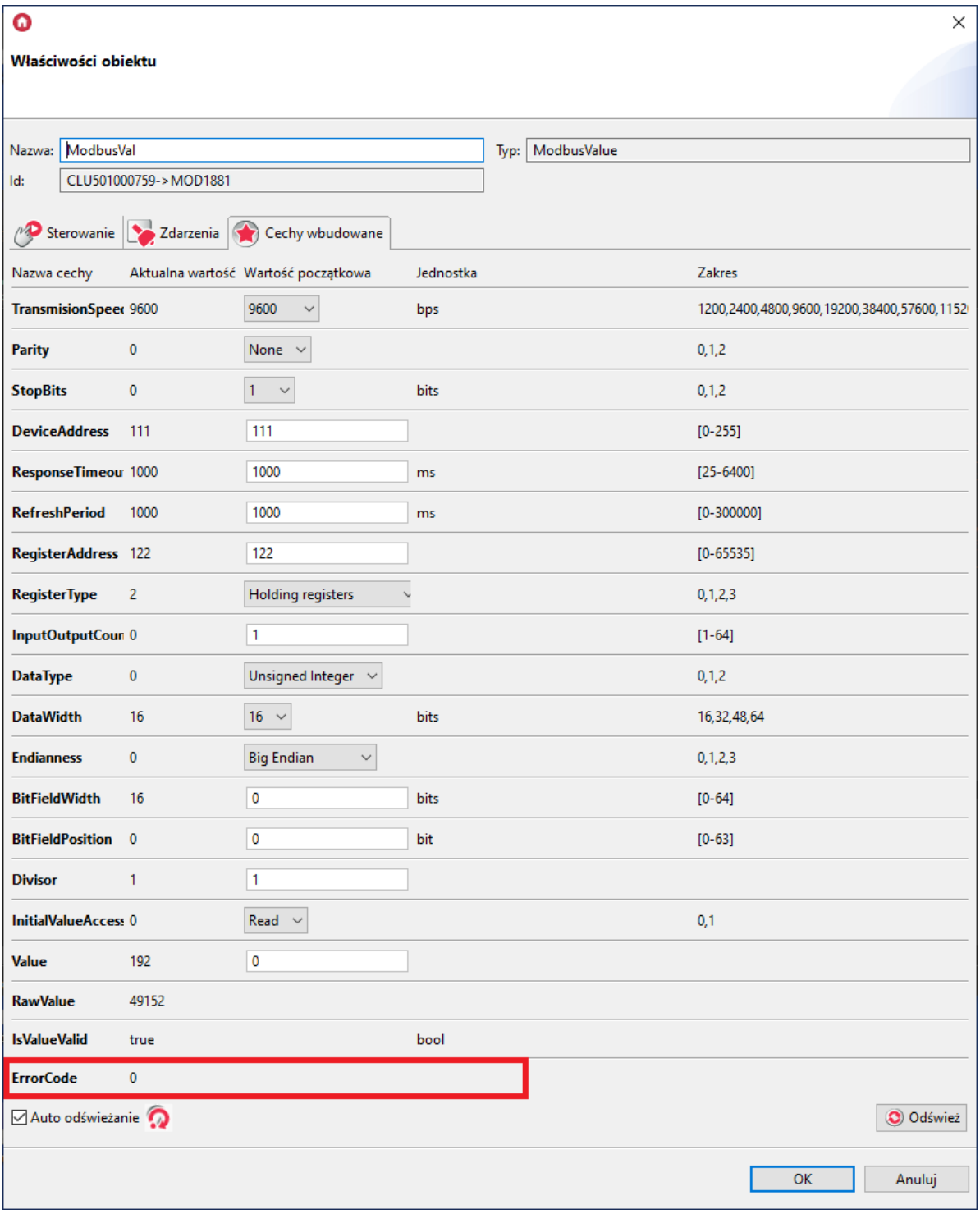
- Przejść do zakładki *Cechy wbudowane* i wprowadzić:
  - **TransmissionSpeed** – prędkość transmisji;
  - **Parity** - parametr ustawia sprawdzanie parzystości: 0 - None 1 - Odd 2 - Even
  - **StopBits** - liczba bitów stopu urządzenia Slave: 0 - 1 bit stopu 1 - 1.5 bitu stopu 2 - 2 bity stopu
  - **DeviceAddress** – adres urządzenia Slave;
  - **ResponseTimeout** - czas na odpowiedź w krokach 25ms;
  - **RefreshPeriod** - minimalny okres odświeżania w krokach 5ms - wartość 0 wyłącza automatyczne odświeżanie;
  - **RegisterAddress** – adres obsługiwanego rejestru;
  - **RegisterType** - typ ustawionego rejestru: 0 - wyjścia dwustanowe 1 - wejścia dwustanowe 2 - rejestry pamiętające 3 - rejestry wejściowe
  - **Divisor** - dzielnik (skala) wartości;
  - **InitialValueAccess** - metoda pierwszego dostępu do wartości Value: 0 - wartość początkowa Value jest odczytywana z urządzenia; 1 - wartość początkowa Value jest zapisywana do urządzenia;

    > **Uwaga!**
    >
    > Wpisanie wybranej wartości w polu `Value` przy ustawieniu cechy `InitialValueAccess` =1 powoduje zapisanie wartości (wysłanie odpowiedniej ramki) po wysłaniu konfiguracji na CLU. Jeśli wartość cechy `InitialValueAccess` =0 wpisana wartość w polu `Value` jest ignorowana.

  - parametry odpowiednie dla wybranego typu rejestru urządzenia Slave - patrz pkt 3.2.



- Wysłać konfigurację oraz zweryfikować połączenie – zakładka *Cechy wbudowane*, cecha `ErrorCode` =0 (poprawny odczyt/zapis):



3. Parametry rejestrów

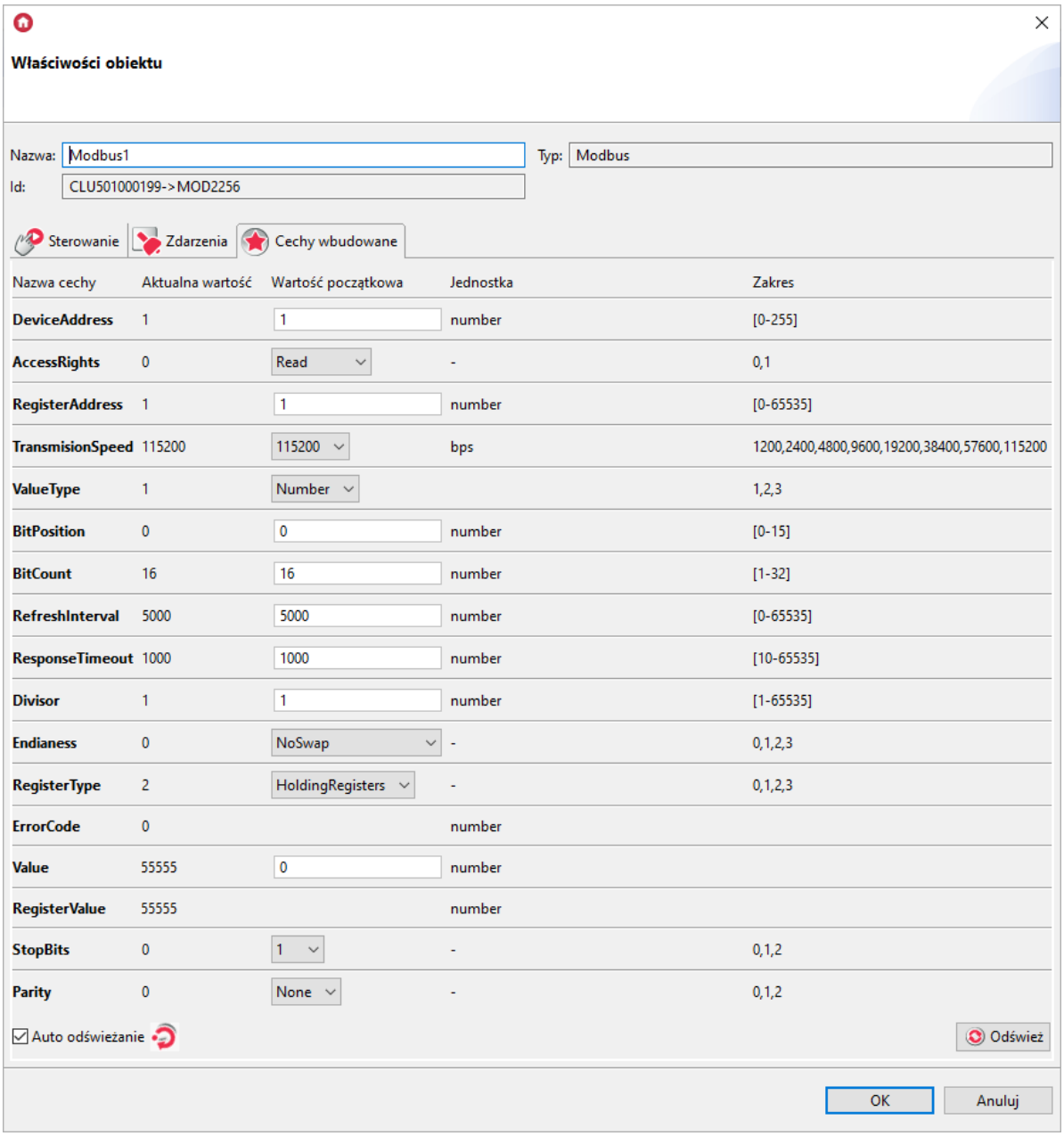
W zależności od typu danego rejestru urządzenia Slave, należy ustawić odpowiednio kolejne dostępne parametry.

3.1. Obiekt wirtualny Modbus

A. Rejestry 16-bitowe

Odczyt 16-bitowych rejestrów pamiętających (`Read Holding Registers`, *FunctionCode=03*):

- `AccessRights`: *Read*;
- `ValueType`: *Number*;
- `BitPosition`: wartość domyślna;
- `BitCount`: 16;
- `Endianness`: wartość domyślna;
- `RegisterType`: *HoldingRegisters*:



Odczyt 16-bitowych rejestrów wejściowych (`Read Input Registers`, *FunctionCode=04*):

- `AccessRights`: *Read*;
- `ValueType`: *Number*;
- `BitPosition`: wartość domyślna;
- `BitCount`: 16;
- `Endianness`: wartość domyślna;
- `RegisterType`: *InputRegisters*.

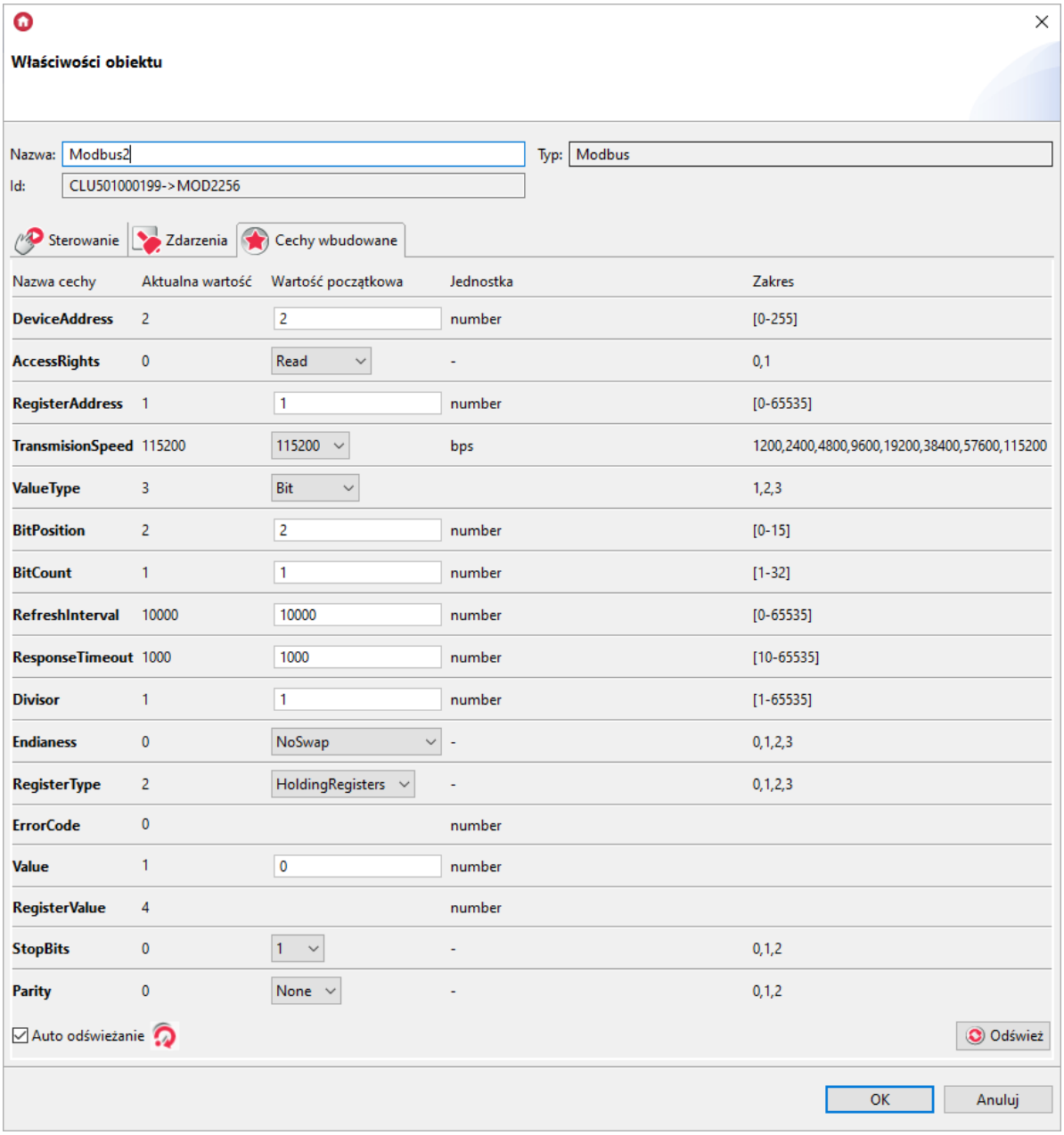
Zapis 16-bitowych rejestrów pamiętających (`Preset/Write Single Holding Register`, *FunctionCode=06*):

- `AccessRights`: *ReadWrite*;
- `ValueType`: *Number*;
- `BitPosition`: wartość domyślna;
- `BitCount`: 16;
- `Endianness`: wartość domyślna;
- `RegisterType`: *HoldingRegisters*.

B. Pola w rejestrach 16-bitowych

Odczyt pól bitowych w 16-bitowym rejestrze pamiętającym (`Read Holding Registers`, *FunctionCode=03*):

- `AccessRights`: *Read*;
- `ValueType`: *Bit*;
- `BitPosition`: 0-15 (pozycja pierwszego interesującego bitu);
- `BitCount`: 1-16 (ilość odczytywanych kolejno bitów);
- `Endianness`: wartość domyślna;
- `RegisterType`: *HoldingRegisters*:



Odczyt pól bitowych w 16-bitowym rejestrze wejściowym (`Read Input Registers`, *FunctionCode=04*):

- `AccessRights`: *Read*;
- `ValueType`: *Bit*;
- `BitPosition`: 0-15 (pozycja pierwszego interesującego bitu);
- `BitCount`: 1-16 (ilość odczytywanych kolejno bitów);
- `Endianness`: wartość domyślna;
- `RegisterType`: *InputRegisters*:

Zapis pól bitowych w 16-bitowym rejestrze pamiętającym (`Preset/Write Single Holding Register`, *FunctionCode=06*):

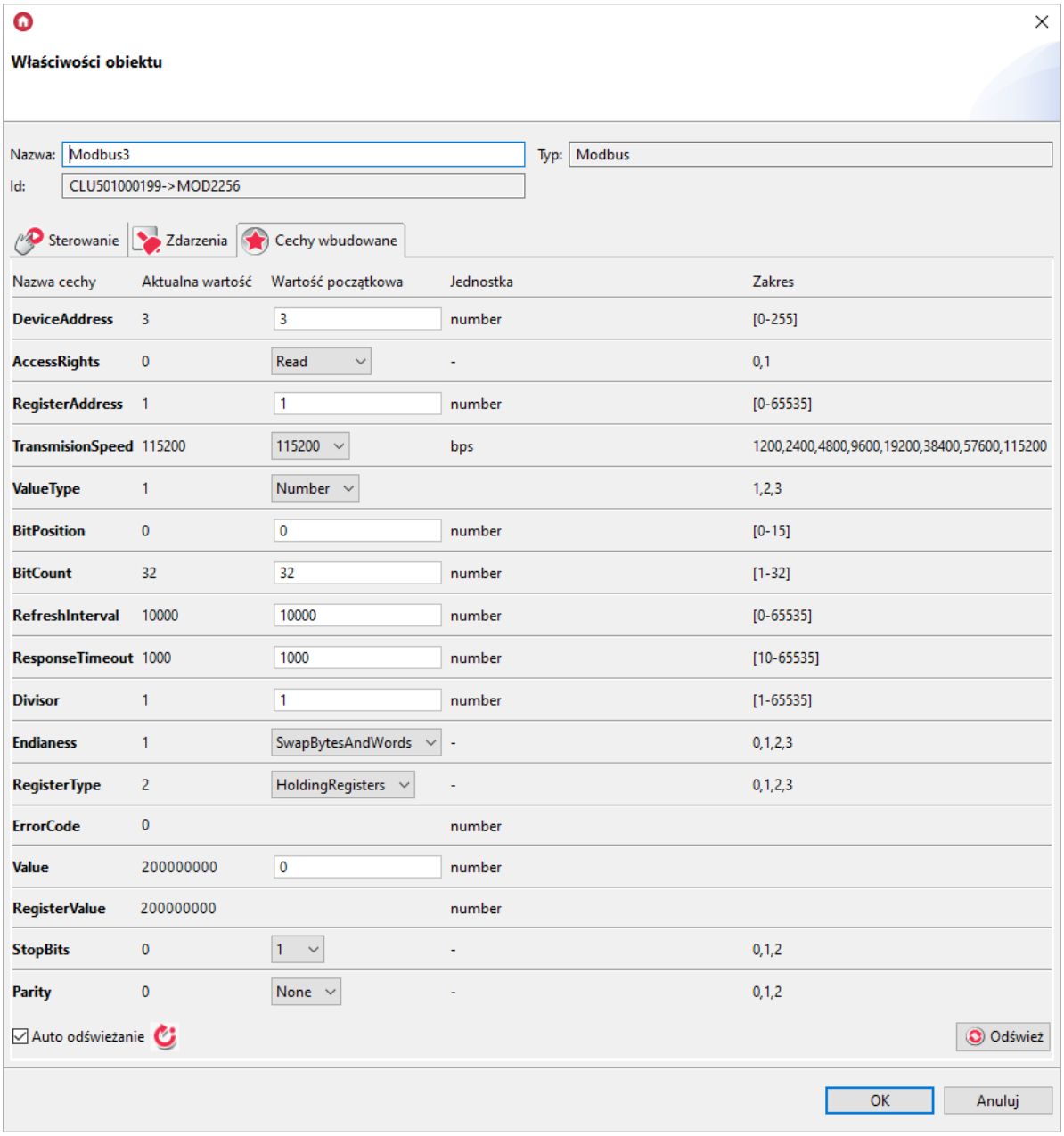
- `AccessRights`: *ReadWrite*;
- `ValueType`: *Bit*;
- `BitPosition`: 0-15 (pozycja pierwszego interesującego bitu);
- `BitCount`: 1-16 (ilość odczytywanych kolejno bitów);

- `Endianness`: wartość domyślna;
- `RegisterType`: *HoldingRegisters*:

C. 32-bitowe wartości całkowite rejestrów

Odczyt 32-bitowych wartości całkowitych rejestru pamiętającego (`Read Holding Registers`, *FunctionCode=03*):

- `AccessRights`: *Read*;
- `ValueType`: *Number*;
- `BitPosition`: wartość domyślna;
- `BitCount`: 32;
- `Endianness`: w przypadku 32-bitowych rejestrów urządzenia Slave najczęściej wymagają zmiany kolejności bajtów i słów - *SwapBytesAndWords*; szczegółowe informacje należy odnaleźć w karcie urządzenia Slave;
- `RegisterType`: *HoldingRegisters*:



Odczyt 32-bitowych wartości całkowitych rejestru wejściowego (`Read Input Registers`, *FunctionCode=04*):

- `AccessRights`: *Read*;
- `ValueType`: *Number*;
- `BitPosition`: wartość domyślna;
- `BitCount`: 32;
- `Endianness`: w przypadku 32-bitowych rejestrów urządzenia Slave najczęściej wymagają zmiany kolejności bajtów i słów - *SwapBytesAndWords*; szczegółowe informacje należy odnaleźć w karcie urządzenia Slave;
- `RegisterType`: *InputRegisters*.

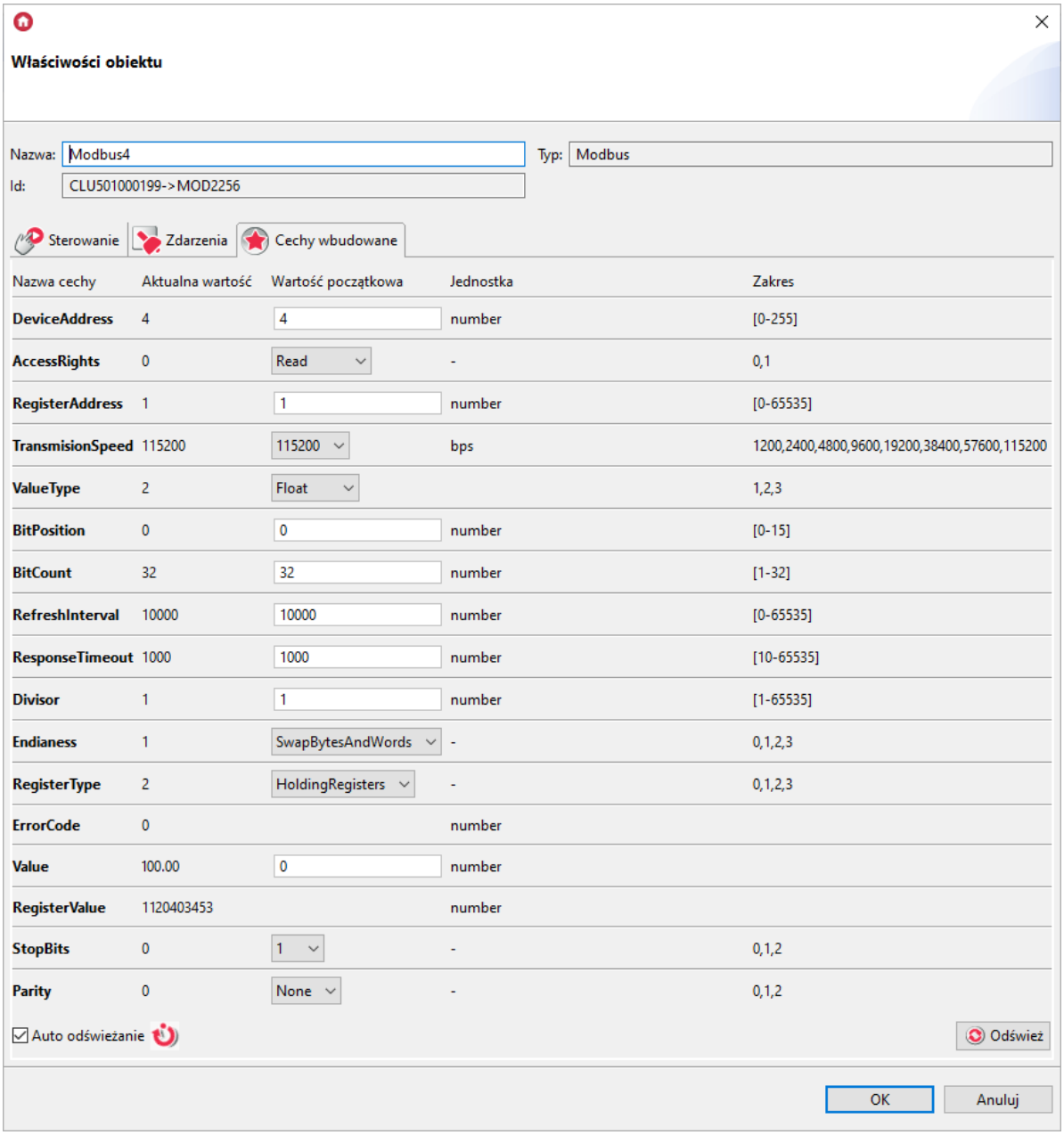
Zapis 32-bitowych wartości całkowitych w rejestrze pamiętającym (`Preset/Write Multiple Holding Registers`, *FunctionCode=16*):

- `AccessRights`: *ReadWrite*;
- `ValueType`: *Number*;
- `BitPosition`: wartość domyślna;
- `BitCount`: 32;
- `Endianness`: w przypadku 32-bitowych rejestrów urządzenia Slave najczęściej wymagają zmiany kolejności bajtów i słów - *SwapBytesAndWords*; szczegółowe informacje należy odnaleźć w karcie urządzenia Slave;
- `RegisterType`: *HoldingRegisters*.

D. 32-bitowe wartości zmiennoprzecinkowe rejestrów

Odczyt 32-bitowych wartości zmiennoprzecinkowych rejestru pamiętającego (`Read Holding Registers`, *FunctionCode=03*):

- `AccessRights`: *Read*;
- `ValueType`: *Float*;
- `BitPosition`: wartość domyślna;
- `BitCount`: 32;
- `Endianness`: w przypadku 32-bitowych rejestrów urządzenia Slave najczęściej wymagają zmiany kolejności bajtów i słów - *SwapBytesAndWords*; szczegółowe informacje należy odnaleźć w karcie urządzenia Slave;
- `RegisterType`: *HoldingRegisters*:



Odczyt 32-bitowych wartości zmiennoprzecinkowych rejestru wejściowego (`Read Input Registers`, *FunctionCode=04*):

- `AccessRights`: *Read*;
- `ValueType`: *Float*;
- `BitPosition`: wartość domyślna;
- `BitCount`: 32;
- `Endianness`: w przypadku 32-bitowych rejestrów urządzenia Slave najczęściej wymagają zmiany kolejności bajtów i słów - *SwapBytesAndWords*; szczegółowe informacje należy odnaleźć w karcie urządzenia Slave;
- `RegisterType`: *InputRegisters*.

Zapis 32-bitowych wartości zmiennoprzecinkowych w rejestrze pamiętającym (`Preset/Write Multiple Holding Registers`, *FunctionCode=16*):

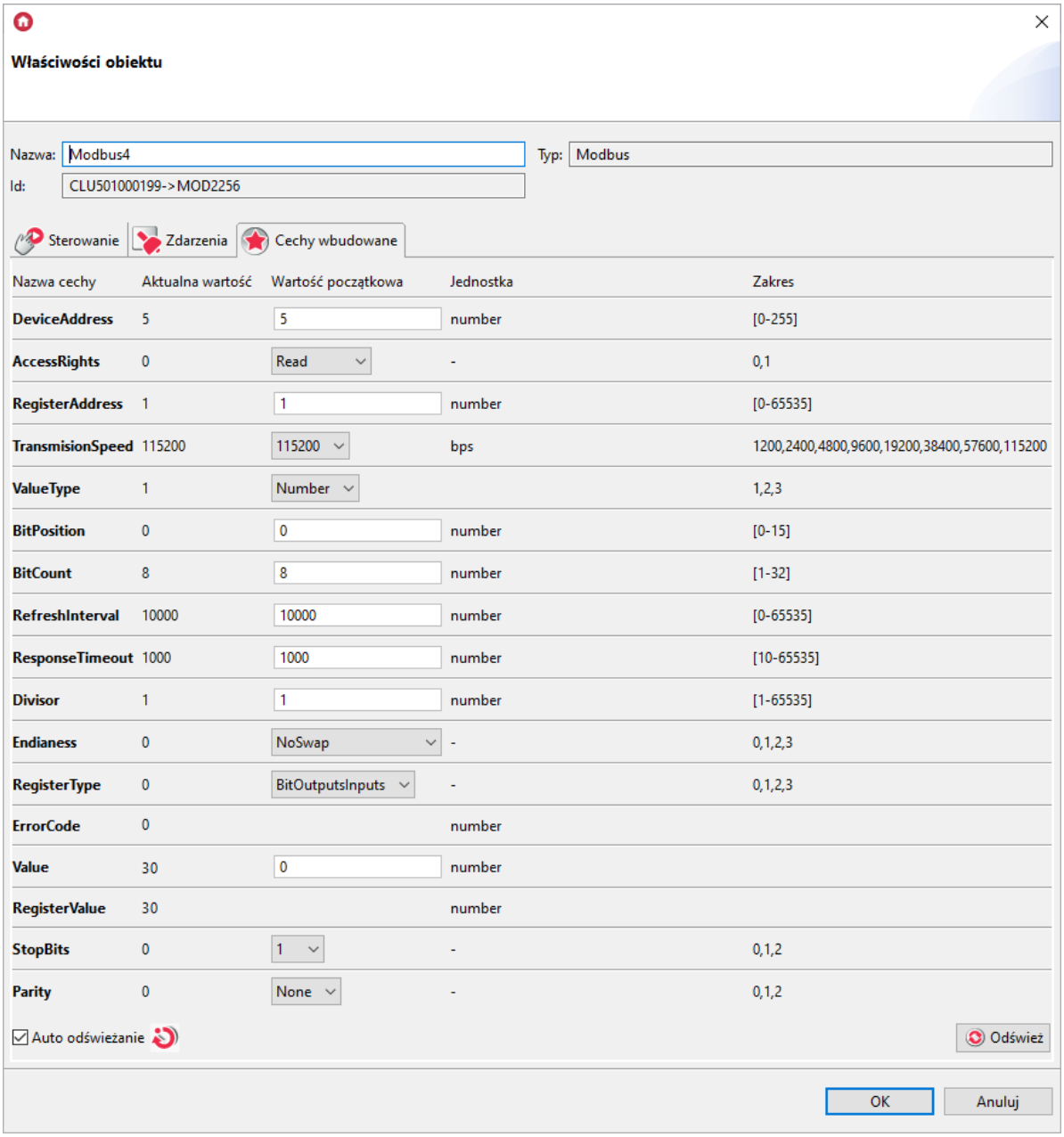
- `AccessRights`: *ReadWrite*;
- `ValueType`: *Float*;

- `BitPosition`: wartość domyślna;
- `BitCount`: 32;
- `Endianness`: w przypadku 32-bitowych rejestrów urządzenia Slave najczęściej wymagają zmiany kolejności bajtów i słów - *Swap bytes and words*; szczegółowe informacje należy odnaleźć w karcie urządzenia Slave;
- `RegisterType`: *HoldingRegisters*.

E. Dyskretne wejścia/wyjścia

Odczyt dyskretnych wyjść/wejść bitowych (`Read Coil Status`, *FunctionCode=01*):

- `AccessRights`: *Read*;
- `ValueType`: *Number*;
- `BitPosition`: wartość domyślna;
- `BitCount`: 1-32;
- `Endianness`: wartość domyślna;
- `RegisterType`: *BitOutputsInputs*:



Odczyt dyskretnych wejść dwustanowych (`Read Discrete Inputs`, *FunctionCode=02*):

- `AccessRights`: *Read*;
- `ValueType`: *Number*;
- `BitPosition`: wartość domyślna;
- `BitCount`: 1-32;
- `Endianness`: wartość domyślna;
- `RegisterType`: *BinaryInputs*.

Zapis dyskretnych wyjść/wejść bitowych (`Force/Write Single Coil`, *FunctionCode=05*; `Force/Write Multiple Coils`, *FunctionCode=15*):

- `AccessRights`: *ReadWrite*;
- `ValueType`: *Number*;
- `BitPosition`: wartość domyślna;
- `BitCount`: 1-32;
- `Endianness`: wartość domyślna;
- `RegisterType`: *BitOutputsInputs*.

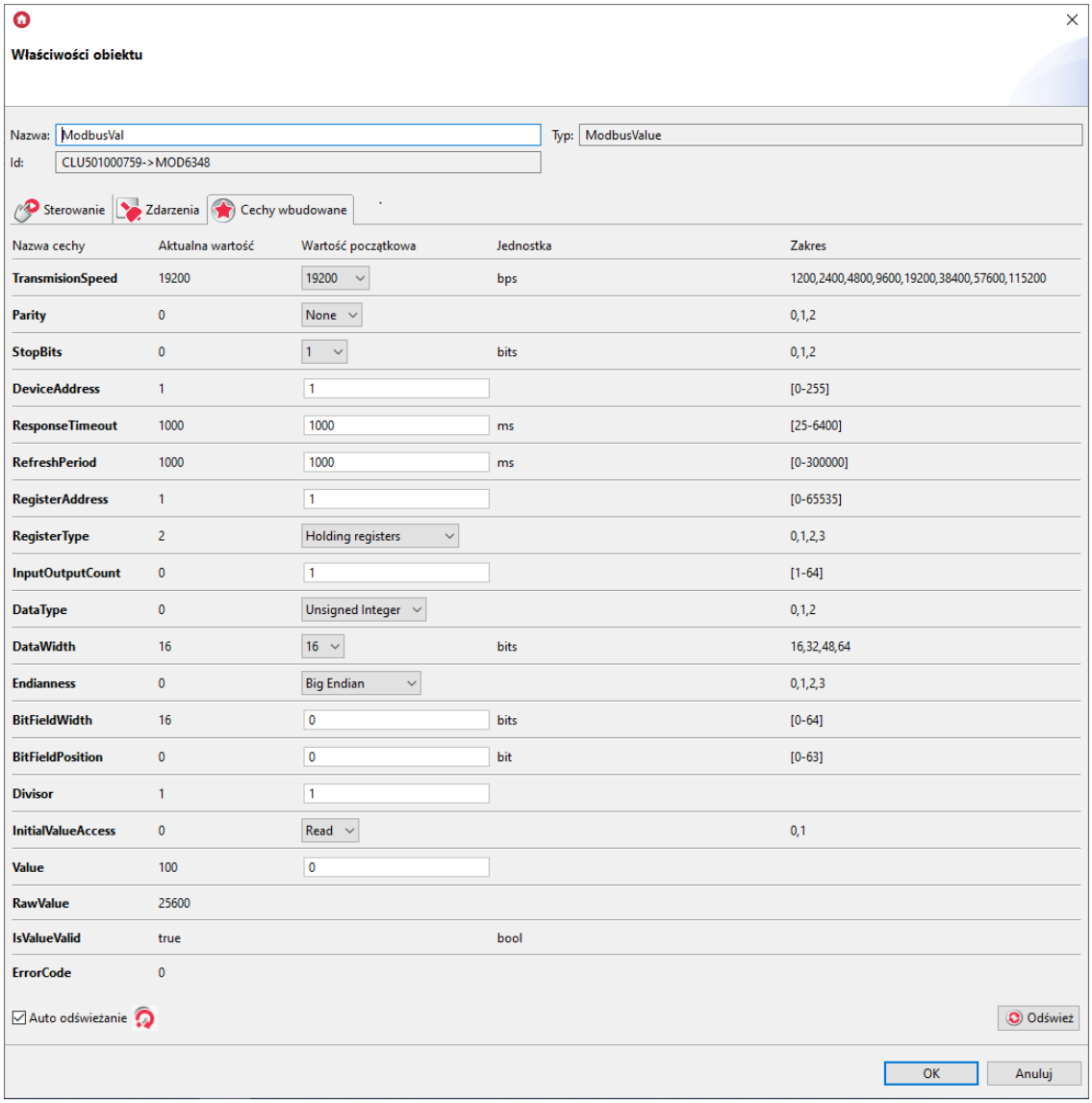
3.2. Obiekt wirtualny ModbusValue

> **Uwaga!** Opisany obiekt wirtualny jest dostępny dla **GRENTON GATE MODBUS MASTER, DIN, Eth (INT-201-E-01)** posiadający **firmware 1.1.10-2140 lub wyższy!**

A. 16-bitowe wartości całkowite rejestrów

Odczyt 16-bitowych rejestrów pamiętających (`Read Holding Registers`, *FunctionCode=03*):

- `RegisterType`: Holding Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 16;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: wartość domyślna;
- `BitFieldPosition`: wartość domyślna.



Odczyt 16-bitowych rejestrów wejściowych (`Read Input Registers`, *FunctionCode=04*):

- `RegisterType`: Input Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 16;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: wartość domyślna;
- `BitFieldPosition`: wartość domyślna.

Zapis 16-bitowych rejestrów pamiętających (`Preset/Write Single Holding Register`, *FunctionCode=06*):

- `RegisterType`: Holding Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 16;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: wartość domyślna;
- `BitFieldPosition`: wartość domyślna.

B. Pola w rejestrach 16-bitowych

Odczyt pól bitowych w 16-bitowym rejestrze pamiętającym (`Read Holding Registers`, *FunctionCode=03*):

- `RegisterType`: Holding Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 16;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: 0 - 16 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = DataWidth);
- `BitFieldPosition`: 0-15 (pozycja pierwszego interesującego bitu).

Uwaga!

Zakres liczbowy cechy `BitFieldWidth` zależy od ustawienia wartości cechy `DataWidth`.

Dla `DataWidth` = 16 zakres cechy `BitFieldWidth` wynosi [0-16].

W chwili ustawienia cechy `BitFieldWidth` na wartość równą 0 i wysłaniu konfiguracji cecha przyjmuje wartość maksymalną dla aktualnie ustawionej wartości cechy `DataWidth`.

Uwaga!

Zakres liczbowy cechy `BitFieldPosition` zależy od ustawienia wartości cechy `DataWidth`.

Dla `DataWidth` = 16 zakres cechy `BitFieldWidth` wynosi [0-15].

Uwaga!

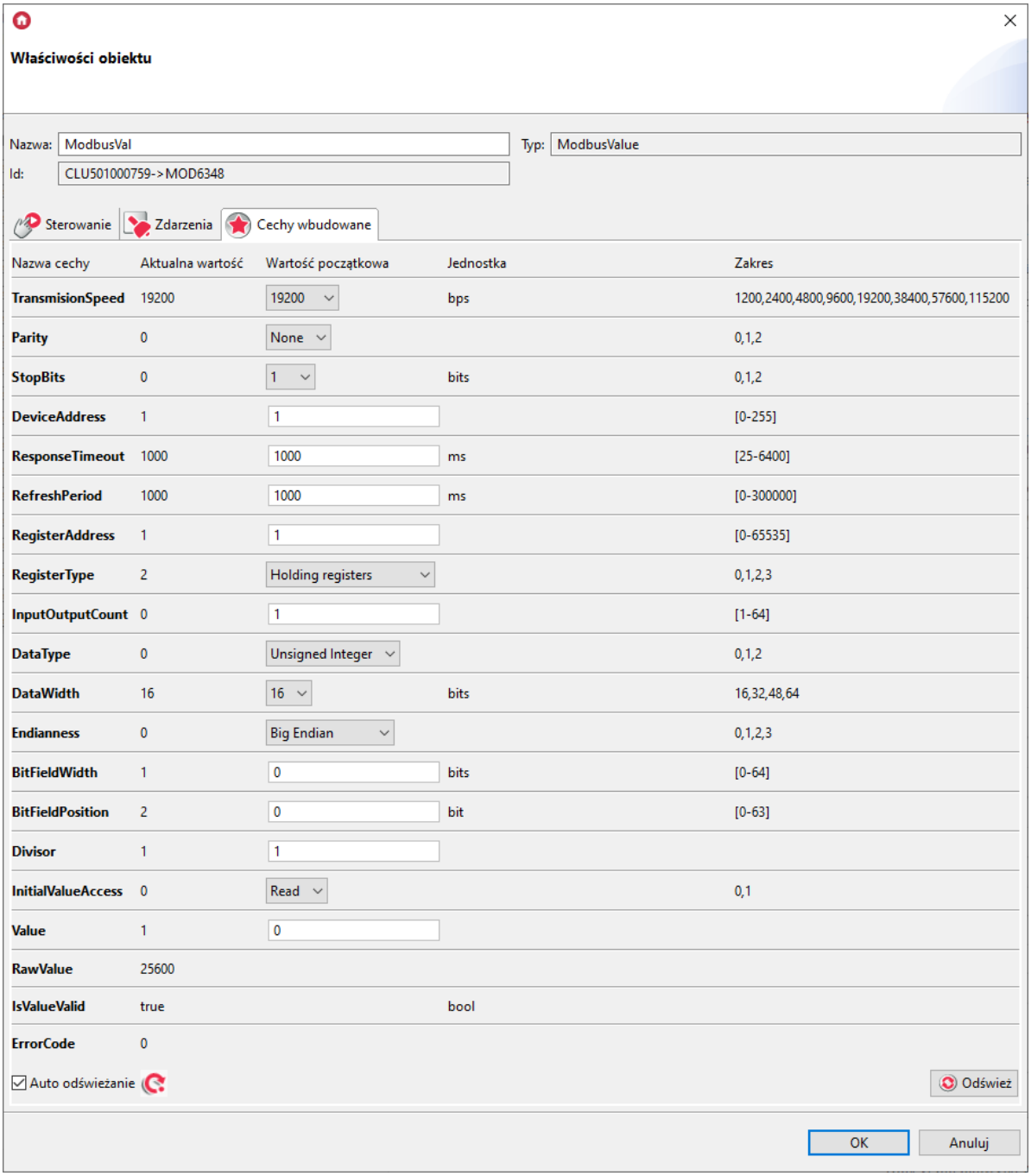
Cechy `BitFieldWidth` i `BitFieldPosition` są uzależnione od cechy `DataWidth`.

Działają one wg zależności: `BitFieldWidth` + `BitFieldPosition` <= `DataWidth`

Na przykład:

W chwili ustawienia `DataWidth` i `BitFieldWidth` = 16 oraz `BitFieldPositon` = 15 cecha `BitFieldWidth` automatycznie przyjmie wartość = 1.

Dla `BitFieldWidth` =0 cecha `BitFieldPosition` przyjmuje zawsze 0.



Odczyt pól bitowych w 16-bitowym rejestrze wejściowym (`Read Input Registers`, *FunctionCode=04*):

- `RegisterType`: Input Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 16;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: 0 - 16 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = DataWidth);
- `BitFieldPosition`: 0-15 (pozycja pierwszego interesującego bitu).

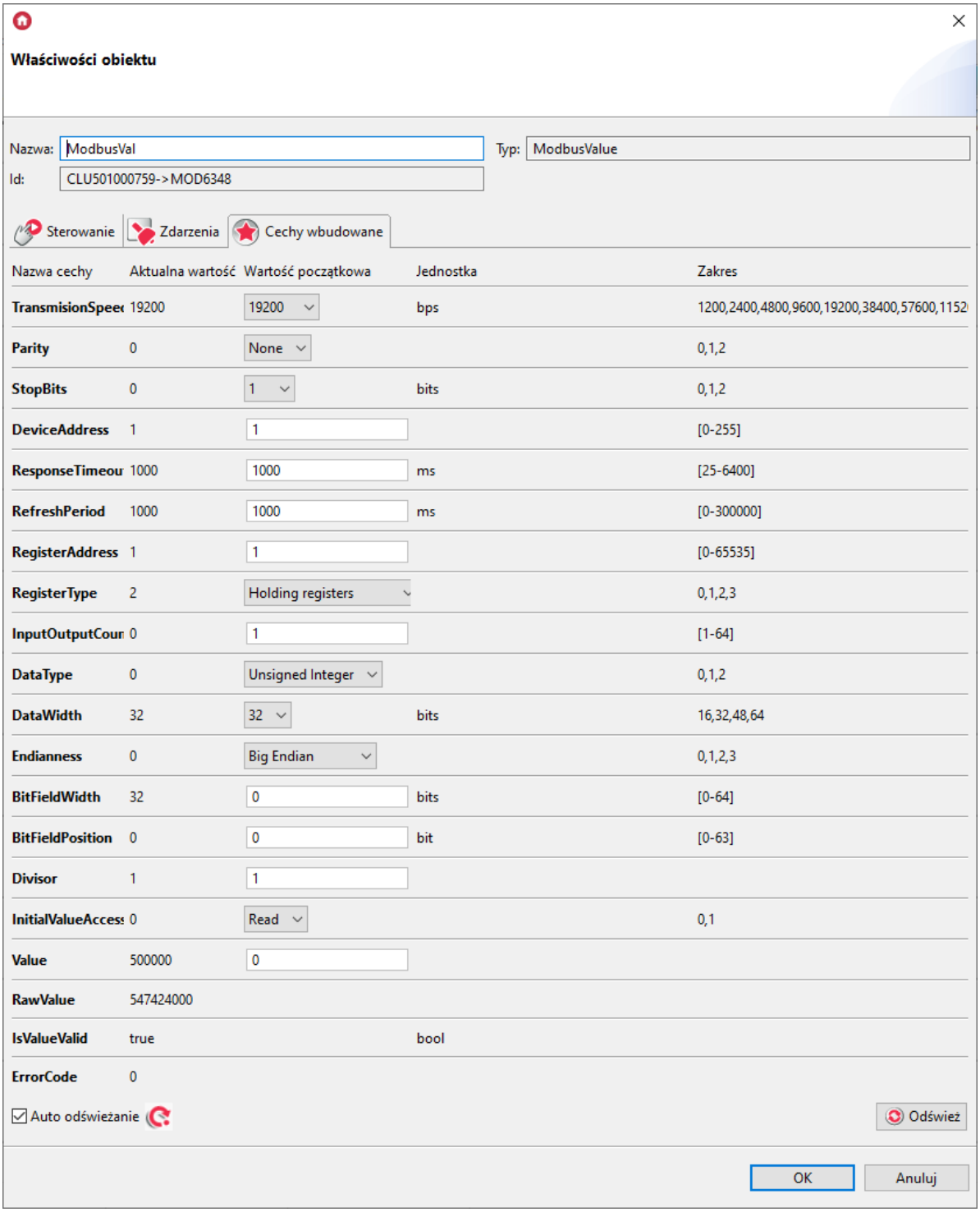
Zapis pól bitowych w 16-bitowym rejestrze pamiętającym (`Preset/Write Single Holding Register`, *FunctionCode=06*):

- `RegisterType`: Holding Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 16;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: 0 - 16 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = DataWidth);
- `BitFieldPosition`: 0-15 (pozycja pierwszego interesującego bitu).

C. 32-bitowe wartości całkowite rejestrów

Odczyt 32-bitowych wartości całkowitych rejestru pamiętającego (`Read Holding Registers`, *FunctionCode=03*):

- `RegisterType`: Holding Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 32;
- `Endianness`: Big Endian;
- `BitFieldWidth`: wartość domyślna;
- `BitFieldPosition`: wartość domyślna.



Odczyt 32-bitowych wartości całkowitych rejestru wejściowego (`Read Input Registers`, *FunctionCode=04*):

- `RegisterType`: Input Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 32;
- `Endianness`: Big Endian;
- `BitFieldWidth`: wartość domyślna;
- `BitFieldPosition`: wartość domyślna.

Zapis 32-bitowych wartości całkowitych w rejestrze pamiętającym (`Preset/Write Multiple Holding Registers`, *FunctionCode=16*):

- `RegisterType`: Holding Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 32;
- `Endianness`: Big Endian;
- `BitFieldWidth`: wartość domyślna;
- `BitFieldPosition`: wartość domyślna.

D. Pola w rejestrach 32-bitowych

Odczyt pól bitowych w 32-bitowym rejestrze pamiętającym (`Read Holding Registers`, *FunctionCode=03*):

- `RegisterType`: Holding Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 32;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: 0-32 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = DataWidth);
- `BitFieldPosition`: 0-31 (pozycja pierwszego interesującego bitu).

Uwaga!

Zakres liczbowy cechy `BitFieldWidth` zależy od ustawienia wartości cechy `DataWidth`.

Dla `DataWidth` = 32 zakres cechy `BitFieldWidth` wynosi [0-32].

W chwili ustawienia cechy `BitFieldWidth` na wartość równą 0 i wysłaniu konfiguracji cecha przyjmuje wartość maksymalną dla aktualnie ustawionej wartości cechy `DataWidth`.

Uwaga!

Zakres liczbowy cechy `BitFieldPosition` zależy od ustawienia wartości cechy `DataWidth`.

Dla `DataWidth` = 32 zakres cechy `BitFieldWidth` wynosi [0-31].

Uwaga!

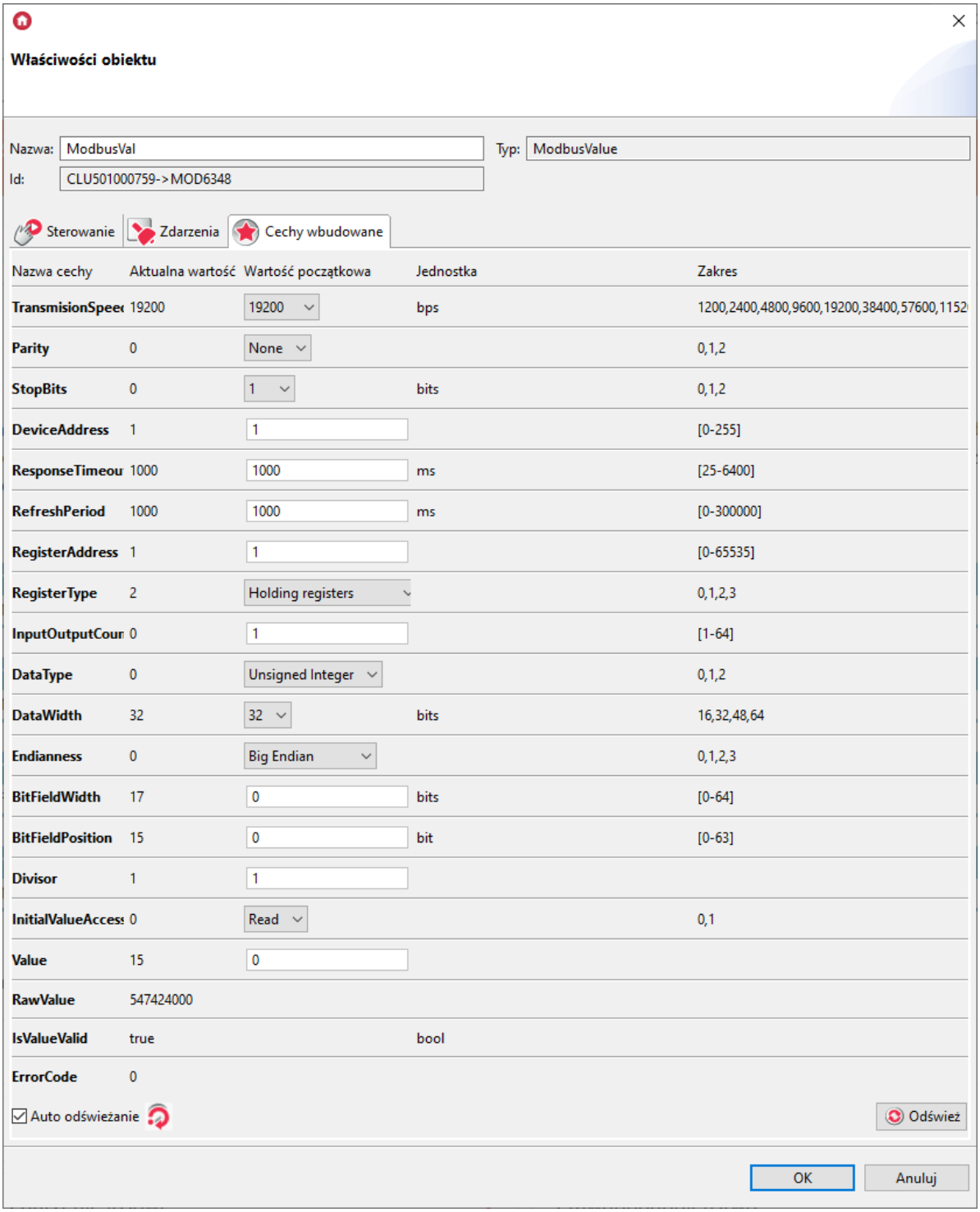
Cechy `BitFieldWidth` i `BitFieldPosition` są uzależnione od cechy `DataWidth`.

Działają one wg zależności: `BitFieldWidth` + `BitFieldPosition` <= `DataWidth`

Na przykład:

W chwili ustawienia `DataWidth` i `BitFieldWidth` = 32 oraz `BitFieldPositon` = 15 cecha `BitFieldWidth` automatycznie przyjmie wartość = 17.

Dla `BitFieldWidth` =0 cecha `BitFieldPosition` przyjmuje zawsze 0.



Odczyt pól bitowych w 32-bitowym rejestrze wejściowym (`Read Input Registers`, *FunctionCode=04*):

- `RegisterType`: Input Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 32;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: 0-32 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = DataWidth);
- `BitFieldPosition`: 0-31 (pozycja pierwszego interesującego bitu).

Zapis pól bitowych w 16-bitowym rejestrze pamiętającym (`Preset/Write Single Holding Register`, *FunctionCode=06*):

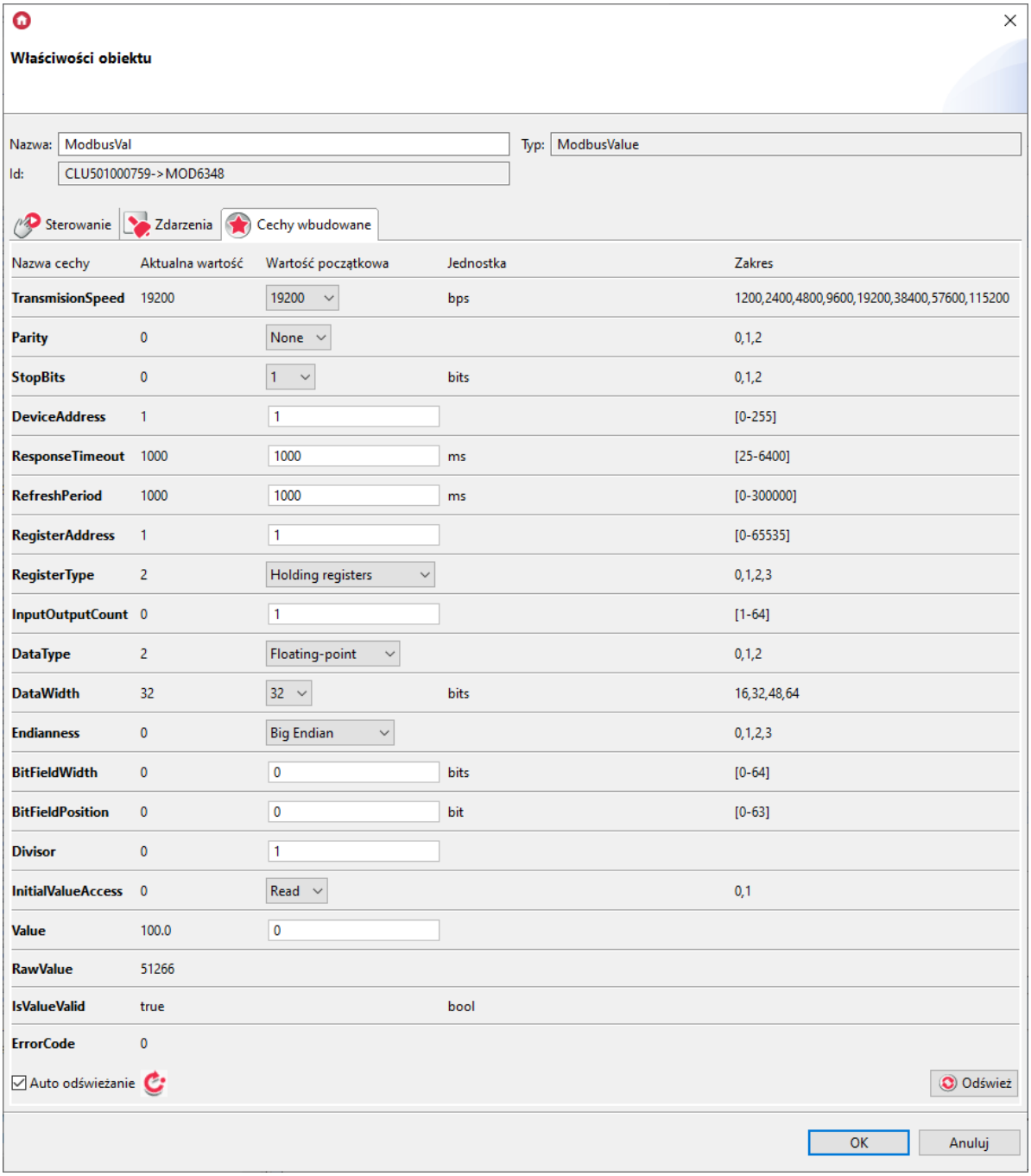
- `RegisterType`: Holding Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 32;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: 0-32 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = DataWidth);
- `BitFieldPosition`: 0-31 (pozycja pierwszego interesującego bitu).

E. 32-bitowe wartości zmiennoprzecinkowe rejestrów

Odczyt 32-bitowych wartości zmiennoprzecinkowych rejestru pamiętającego (`Read Holding Registers`, *FunctionCode=03*):

- `RegisterType`: Holding Registers;
- `DataType`: Floating-point;
- `DataWidth`: 32;
- `Endianness`: Big Endian;

> Dla `DataType` = Floating-point cechy `BitFieldWidth`, `BitFieldPosition` oraz `Divisor` są nieaktywne i przyjmują zawsze wartość 0!



Odczyt 32-bitowych wartości zmiennoprzecinkowych rejestru wejściowego (`Read Input Registers`, *FunctionCode=04*):

- `RegisterType`: Input Registers;
- `DataType`: Floating-point;
- `DataWidth`: 32;
- `Endianness`: Big Endian.

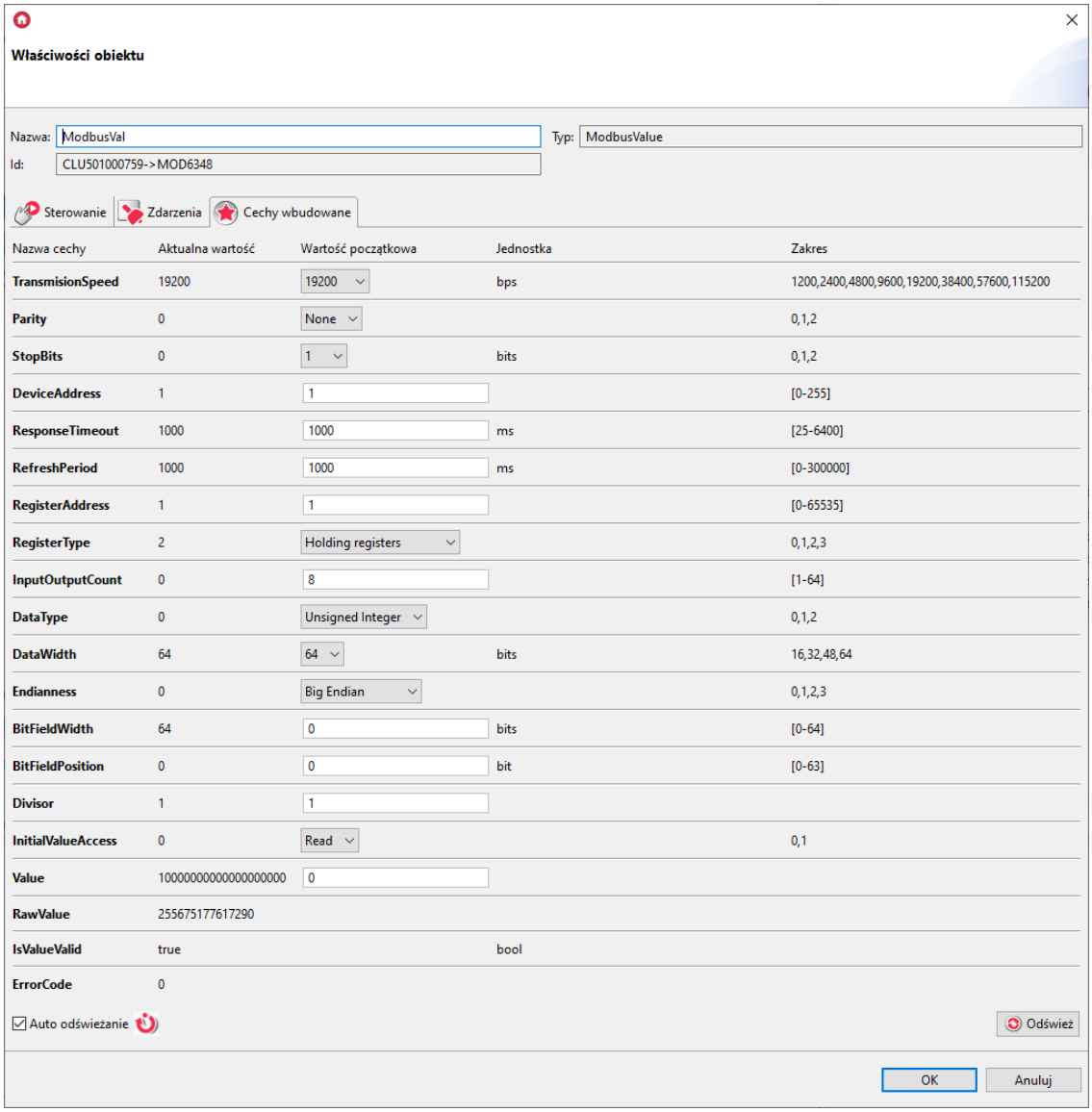
Zapis 32-bitowych wartości zmiennoprzecinkowych w rejestrze pamiętającym (`Preset/Write Multiple Holding Registers`, *FunctionCode=16*):

- `RegisterType`: Holding Registers;
- `DataType`: Floating-point;
- `DataWidth`: 32;
- `Endianness`: Big Endian.

F. 64-bitowe wartości całkowite rejestrów

Odczyt 64-bitowych rejestrów pamiętających (`Read Holding Registers` , *FunctionCode=03*):

- `RegisterType` : Holding Registers;
- `DataType` : Unsigned Integer;
- `DataWidth` : 64
- `Endianness` : wartość domyślna;
- `BitFieldWidth` : wartość domyślna;
- `BitFieldPosition` : wartość domyślna.



Odczyt 64-bitowych rejestrów wejściowych (`Read Input Registers` , *FunctionCode=04*):

- `RegisterType` : Input Registers;
- `DataType` : Unsigned Integer;
- `DataWidth` : 64
- `Endianness` : wartość domyślna;
- `BitFieldWidth` : wartość domyślna;
- `BitFieldPosition` : wartość domyślna.

Zapis 64-bitowych rejestrów pamiętających (`Preset/Write Single Holding Register`, *FunctionCode=06*):

- `RegisterType`: Holding Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 64;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: wartość domyślna;
- `BitFieldPosition`: wartość domyślna.

G. Pola w rejestrach 64-bitowych

Odczyt pól bitowych w 64-bitowym rejestrze pamiętającym (`Read Holding Registers`, *FunctionCode=03*):

- `RegisterType`: Holding Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 64;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: 0-64 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = DataWidth);
- `BitFieldPosition`: 0-63 (pozycja pierwszego interesującego bitu).

Uwaga!

Zakres liczbowy cechy `BitFieldWidth` zależy od ustawienia wartości cechy `DataWidth`.

Dla `DataWidth` = 64 zakres cechy `BitFieldWidth` wynosi [0-64].

W chwili ustawienia cechy `BitFieldWidth` na wartość równą 0 i wysłaniu konfiguracji cecha przyjmuje wartość maksymalną dla aktualnie ustawionej wartości cechy `DataWidth`.

Uwaga!

Zakres liczbowy cechy `BitFieldPosition` zależy od ustawienia wartości cechy `DataWidth`.

Dla `DataWidth` = 64 zakres cechy `BitFieldPosition` wynosi [0-63].

Uwaga!

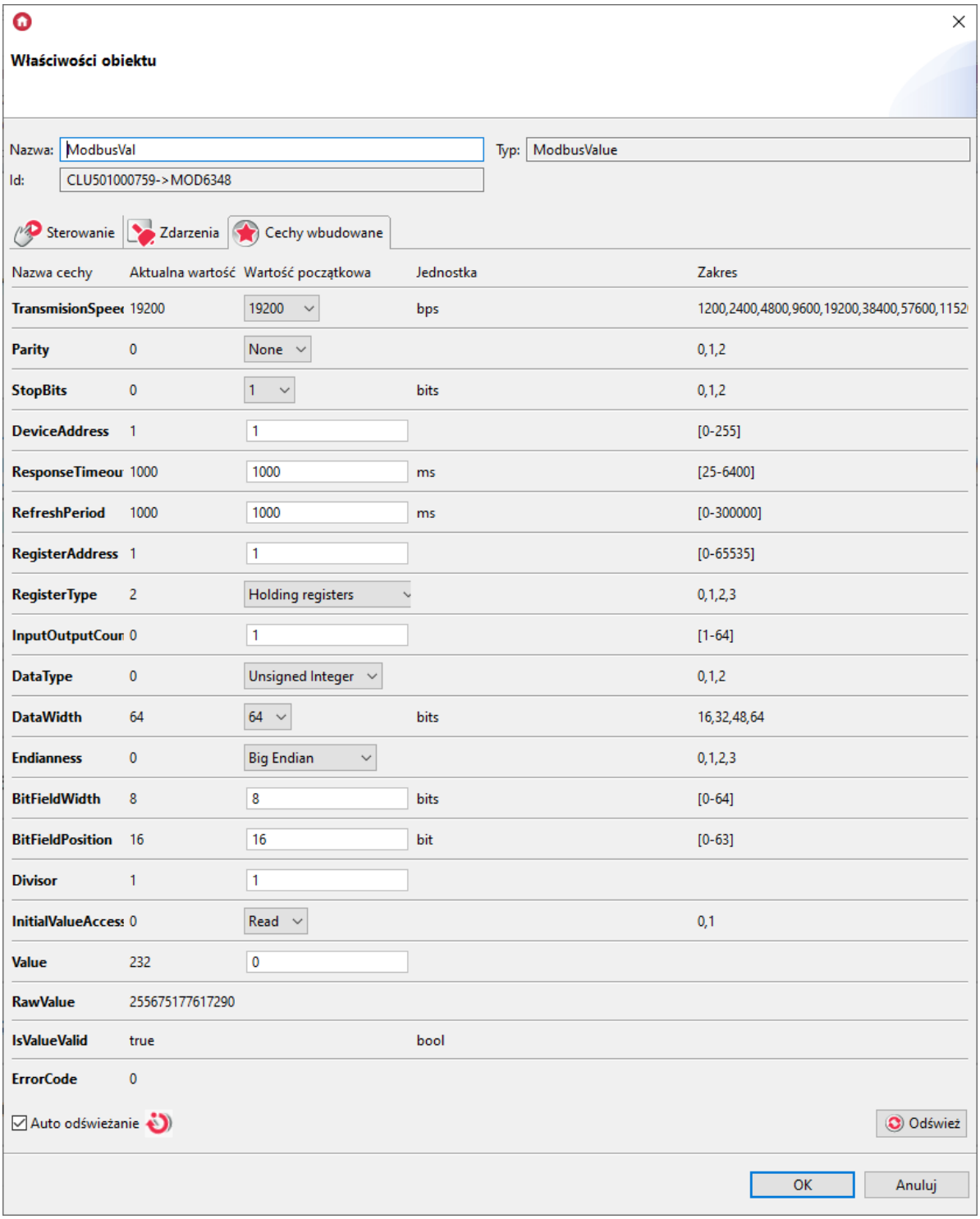
Cechy `BitFieldWidth` i `BitFieldPosition` są uzależnione od cechy `DataWidth`.

Działają one wg zależności: `BitFieldWidth` + `BitFieldPosition` <= `DataWidth`

Na przykład:

W chwili ustawienia `DataWidth` i `BitFieldWidth` = 60 oraz `BitFieldPositon` = 15 cecha `BitFieldWidth` automatycznie przyjmie wartość = 49

Dla `BitFieldWidth` =0 cecha `BitFieldPosition` przyjmuje zawsze 0.



Odczyt pól bitowych w 64-bitowym rejestrze wejściowym (`Read Input Registers`, *FunctionCode=04*):

- `RegisterType`: Input Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 64;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: 0-64 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = DataWidth);
- `BitFieldPosition`: 0-63 (pozycja pierwszego interesującego bitu).

Zapis pól bitowych w 16-bitowym rejestrze pamiętającym (`Preset/Write Single Holding Register`, *FunctionCode=06*):

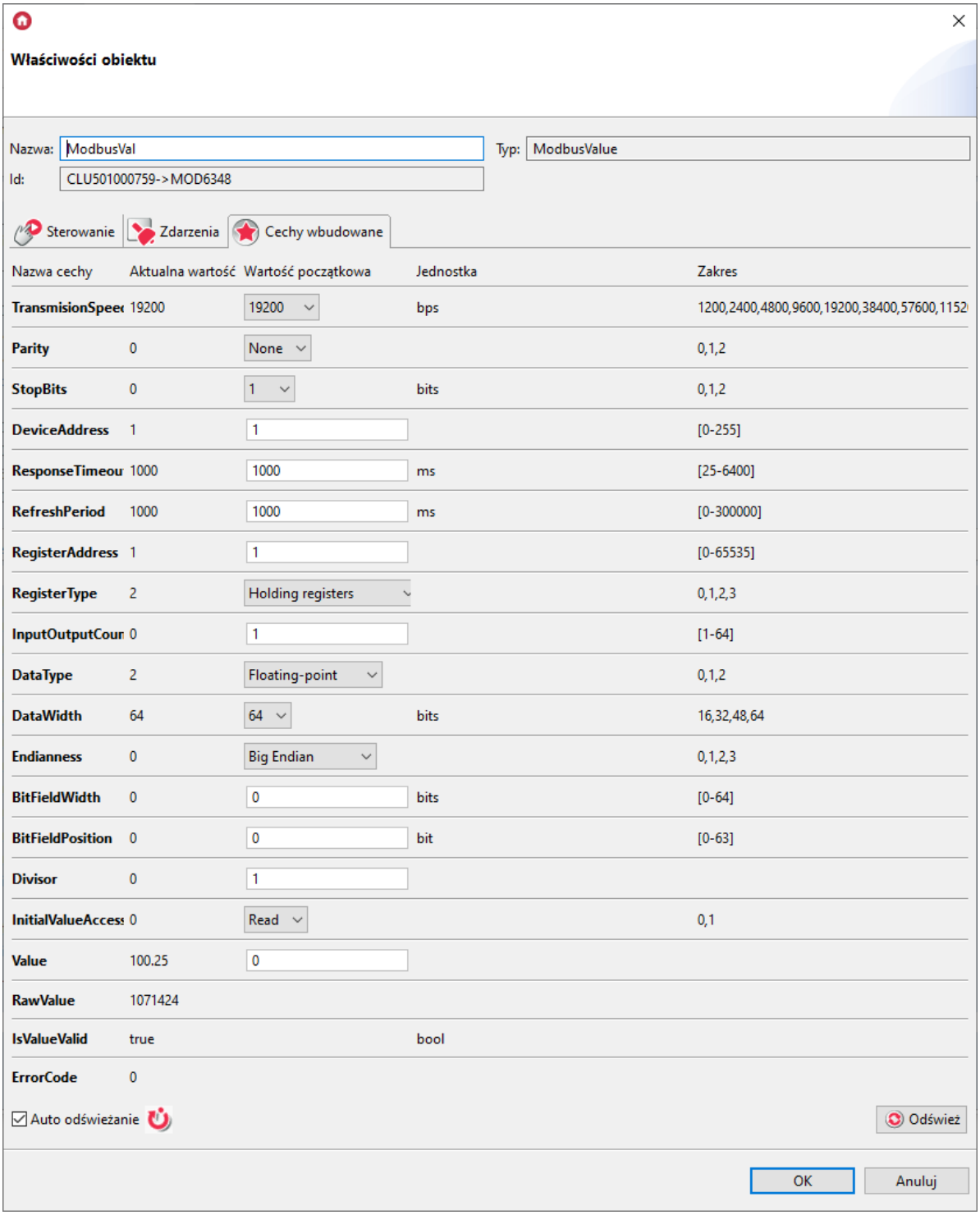
- `RegisterType`: Holding Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 64;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: 0-64 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = DataWidth);
- `BitFieldPosition`: 0-63 (pozycja pierwszego interesującego bitu).

H. 64-bitowe wartości zmiennoprzecinkowe rejestrów

Odczyt 64-bitowych wartości zmiennoprzecinkowych rejestru pamiętającego (`Read Holding Registers`, *FunctionCode=03*):

- `RegisterType`: Holding Registers;
- `DataType`: Floating-point;
- `DataWidth`: 64;
- `Endianness`: Big Endian.

> Dla `DataType` = Floating-point cechy `BitFieldWidth`, `BitFieldPosition` oraz `Divisor` są nieaktywne i przyjmują zawsze wartość 0!



Odczyt 64-bitowych wartości zmiennoprzecinkowych rejestru wejściowego (`Read Input Registers`, *FunctionCode=04*):

- `RegisterType`: Input Registers;
- `DataType`: Floating-point;
- `DataWidth`: 64;
- `Endianness`: Big Endian.

Zapis 64-bitowych wartości zmiennoprzecinkowych w rejestrze pamiętającym (`Preset/Write Multiple Holding Registers`, *FunctionCode=16*):

- `RegisterType`: Holding Registers;

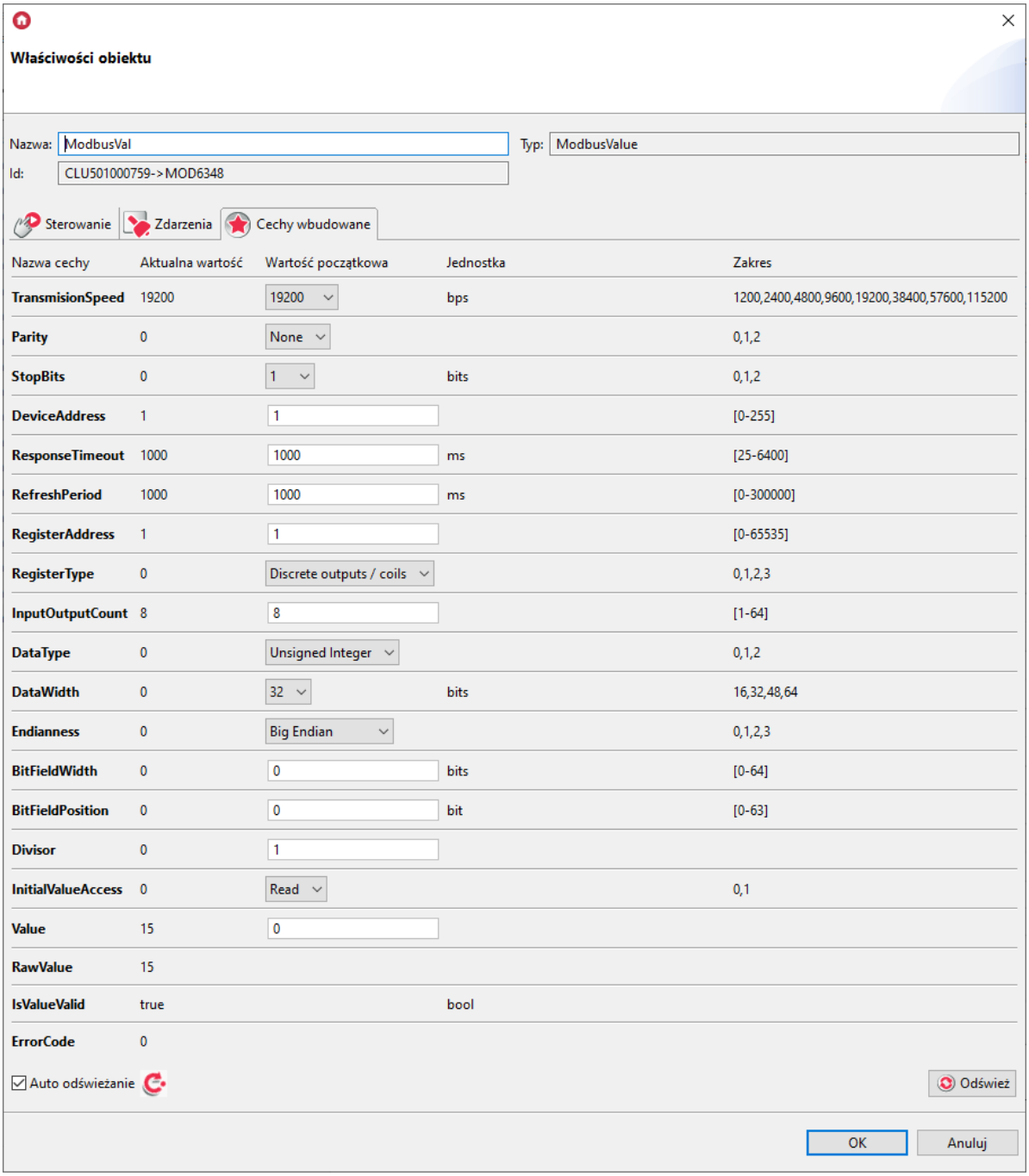
- `DataType`: Floating-point;
- `DataWidth`: 64;
- `Endianness`: Big Endian.

J. Dyskretne wejścia/wyjścia

Odczyt dyskretnych wyjść/wejść bitowych (`Read Coil Status`, *FunctionCode=01*):

- `RegisterType`: Discrete outputs / coils;
- `InputOutputCount`: 1-64 (liczba dyskretnych wejść/wyjść podlegających operacji odczytu/zapisu).

> Dla typów rejestrów *Discrete outputs / coils* i *Discrete Inputs* cechy `DataType`, `DataWidth`, `Endianness`, `BitFieldWidth`, `BitFieldPosition`, `Divisor` są nieaktywne i przyjmują wartość 0!



Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
TransmisionSpeed	19200	19200	bps	1200,2400,4800,9600,19200,38400,57600,115200
Parity	0	None		0,1,2
StopBits	0	1	bits	0,1,2
DeviceAddress	1	1		[0-255]
ResponseTimeout	1000	1000	ms	[25-6400]
RefreshPeriod	1000	1000	ms	[0-300000]
RegisterAddress	1	1		[0-65535]
RegisterType	0	Discrete outputs / coils		0,1,2,3
InputOutputCount	8	8		[1-64]
DataType	0	Unsigned Integer		0,1,2
DataWidth	0	32	bits	16,32,48,64
Endianness	0	Big Endian		0,1,2,3
BitFieldWidth	0	0	bits	[0-64]
BitFieldPosition	0	0	bit	[0-63]
Divisor	0	1		
InitialValueAccess	0	Read		0,1
Value	15	0		
RawValue	15			
IsValueValid	true		bool	
ErrorCode	0			

Odczyt dyskretnych wejść dwustanowych (`Read Discrete Inputs`, *FunctionCode=02*):

- `RegisterType`: Discrete Inputs;
- `InputOutputCount`: 1-64 (liczba dyskretnych wejść/wyjść podlegających operacji odczytu/zapisu).

Zapis dyskretnych wyjść/wejść bitowych (`Force/Write Single Coil`, *FunctionCode=05*; `Force/Write Multiple Coils`, *FunctionCode=15*):

- `RegisterType`: Discrete outputs / coils;
- `InputOutputCount`: 1-64 (liczba dyskretnych wejść/wyjść podlegających operacji odczytu/zapisu).

4. Przywracanie ustawień fabrycznych - *Hard Reset*

Uruchomienie funkcji *Hard Reset* na module GATE Modbus powoduje:

- Usunięcie zapisanej konfiguracji;
- Sformatowanie partycji pamięci flash;
- Usunięcie wszystkich utworzonych obiektów LUA;
- Utratę komunikacji pomiędzy OM / HM a modułem Gate.

W celu przywrócenia ustawień fabrycznych funkcją *Hard Reset* należy wykonać następujące czynności (zgodnie z podaną kolejnością):

- Odłączyć zasilanie od modułu Gate;
- Nacisnąć i przytrzymać przycisk *Reset* na module (przycisk znajduje się pod dolną zaślepką modułu);
- Podłączyć zasilanie do modułu Gate;
- Trzymać wciśnięty przycisk *Reset* przez co najmniej 10 sekund - podczas resetu dioda zielona będzie świecić światłem ciągłym. Prawidłowe wykonanie resetu zostanie potwierdzone 3-krotnym mrugnięciem diody zielonej.
- Po upływie 10 sekund zwolnić przycisk *Reset*
- Odczekać około 60 sekund aż do momentu, gdy na module diody - zielona oraz czerwona - będą mrugać naprzemiennie (tryb *Emergency*)

Po wykonaniu procedury na module zostanie wyczyszczona konfiguracja, natomiast sam moduł przestanie być widoczny (brak odpowiedzi na *Keep-Alive*) w projekcie z poziomu Object Managera. Aby ponownie przywrócić moduł, należy wykonać CLU Discovery a następnie wysłać konfigurację.

5. Parametry konfiguracyjne

> **Uwaga!** Opisana funkcjonalność oraz integracja jest dostępna dla **GRENTON GATE MODBUS MASTER, DIN, Eth (INT-201-E-01)** posiadający **firmware 1.1.0-2034C lub wyższy!**

A. Obiekt GATE

CECHY

Nazwa	Opis
`Uptime`	Czas pracy urządzenia od ostatniego resetu (w sekundach)
`FirmwareVersion`	Wersja oprogramowania Gate
`PrimaryDNS`	Preferowany serwer DNS
`SecondaryDNS`	Alternatywny serwer DNS
`ClientReportInterval`	Okres raportowania o zmianach cech
`Date`	Aktualna data
`Time`	Aktualny czas (hh:mm:ss)
`LocalTime`	Aktualny lokalny znacznik czasu
`TimeZone`	Strefa czasowa
`UnixTime`	Aktualny uniksowy znacznik czasu
`UseCloud`	Określa czy GATE łączy się do chmury
`CloudConnection`	Określa status połączenia GATE z chmurą
`NTPTimeout`	Czas oczekiwania na odpowiedź z serwera NTP
`UseNTP`	Określa czy GATE używa NTP

METODY

Nazwa	Opis
`StartConsole`	Uruchamia konsolę Lua
`StartConsoleOnReboot`	Uruchamia konsolę Lua przy kolejnym uruchomieniu modułu
`SetClientReportInterval`	Ustawia okres raportowania o zmianach cech
`SetPrimaryDNS`	Ustawia cechę PrimaryDNS
`SetSecondaryDNS`	Ustawia cechę SecondaryDNS
`SetDateTime`	Ustawia datę i czas

ZDARZENIA:

Nazwa	Opis
`OnInit`	Zdarzenie wywoływane jednorazowo w momencie inicjalizacji urządzenia

B. Obiekt Modbus

CECHY

Nazwa	Opis
`DeviceAddress`	Adres urządzenia Modbus typu Slave
`AccessRights`	Tryb pracy: *read* (0 – odczyt); *read/write* (1 - odczyt/zapis)
`RegisterAddress`	Adres obsługiwanego rejestru
`TransmissionSpeed`	Prędkość transmisji
`ValueType`	Typ zmiennej (1 – *number*; 2 – *float*; 3 – *bit*)
`BitPosition`	Pozycja bitu (dotyczy dostępu bitowego do 16-bitowych rejestrów)
`BitCount`	Liczba bitów rejestru do odczytania
`RefreshInterval`	Czas odświeżania
`ResponseTimeout`	Czas na odpowiedź
`Divisor`	Dzielnik
`Endianness`	Kolejność bajtów i słów [7]: *No swap* (0 – bez zamiany); *Swap bytes and words* (1 - zamiana kolejności bajtów i słów); *Swap bytes* (2 - zamiana kolejności bajtów w obrębie każdego ze słów); *Swap words* (3 - zamiana słów)
`RegisterType`	Typ rejestru Modbus (0 – wejścia/wyjścia bitowe, 1 – wejścia dwustanowe, 2 – rejestry pamiętające, 3 – rejestry wejściowe)
`ErrorCode`	Kod błędu: (**-3** - błąd ramki; **-2** - przekroczenie czasu odpowiedzi; **-1** - nieaktualna wartość ostatniego odczytanego rejestru; **0** – poprawny odczyt/zapis rejestru; **1** - niedozwolona funkcja; **2** - niedozwolony numer rejestru; **3** - niedozwolona wartość danej; **4** - uszkodzenie w przyłączonym urządzeniu; **5** - potwierdzenie pozytywne; **6** - brak gotowości/komunikat usunięty; **7** - potwierdzenie negatywne; **8** - błąd parzystości pamięci)
`Value`	Wartość odczytu/zapisu
`RegisterValue`	Nieprzeskalowana wartość rejestru
`StopBits`	Bity stopu: 0 - 1 bit stopu 1 - 1.5 bitu stopu 2 - 2 bity stopu
`Parity`	Bit parzystości: 0 - None 1 - Odd 2 - Even

METODY

Nazwa	Opis
`SetDeviceAddress`	Ustawia adres urządzenia Modbus typu Slave
`SetAccessRights`	Ustawia tryb pracy: odczyt lub odczyt/zapis
`SetRegisterAddress`	Ustawia adres obsługiwanego rejestru
`SetTransmissionSpeed`	Ustawia prędkość transmisji
`SetValueType`	Ustawia typ zmiennej
`SetBitPosition`	Ustawia pozycję bitu
`SetBitCount`	Ustawia liczbę bitów rejestru do odczytania
`SetRefreshInterval`	Ustawia czas odświeżania
`SetResponseTimeout`	Ustawia czas oczekiwania na odpowiedź
`SetDivisor`	Ustawia dzielnik
`SetEndianness`	Ustawia typ kolejności bajtów
`SetRegisterType`	Ustawia typ rejestru Modbus
`SetValue`	Ustawia wartość odczytu/zapisu
`SetStopBits`	Ustawia liczbę bitów stopu
`SetParity`	Ustawia sprawdzanie parzystości

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane przy zmianie stanu (niezależnie od wartości)
`OnError`	Zdarzenie wywołane, gdy urządzenie slave zgłasza błąd

C. Obiekt ModbusValue

> **Uwaga!** Opisany obiekt wirtualny jest dostępny dla **GRENTON GATE MODBUS MASTER, DIN, Eth (INT-201-E-01)** posiadający **firmware 1.1.10-2140 lub wyższy!**

CECHY

Nazwa	Opis
`TransmissionSpeed`	Prędkość transmisji
`Parity`	Bit parzystości: 0 - None 1 - Odd 2 - Even
`StopBits`	Bity stopu: 0 - 1 bit stopu 1 - 1.5 bitu stopu 2 - 2 bity stopu
`DeviceAddress`	Adres urządzenia Slave Modbus
`ResponseTimeout`	Czas na odpowiedź w krokach 25ms
`RefreshPeriod`	Minimalny okres odświeżania w krokach 5ms. Wartość 0 wyłącza automatyczne odświeżanie
`RegisterAddress`	Adres obsługiwanego rejestru
`RegisterType`	Typ rejestru Modbus (0 – wejścia/wyjścia bitowe, 1 – wejścia dwustanowe, 2 – rejestry pamiętające, 3 – rejestry wejściowe)
`InputOutputCount`	Określa liczbę dyskretnych wejść/wyjść podlegających operacji odczytu/zapisu
`DataType`	Typ wartości: 0 - Liczba całkowita, stałoprzecinkowa lub pole bitowe bez bitu znaku 1 - Liczba całkowita, stałoprzecinkowa lub pole bitowe z bitem znaku 2 - Liczba zmiennoprzecinkowa
`DataWidth`	Szerokość danych (od 1 do 4 rejestrów 16 bitowych)
`Endianness`	Kolejność bajtów: 0 - kolejność słów: Big Endian; kolejność bajtów w słowie: Big Endian 1 - kolejność słów: Little Endian; kolejność bajtów w słowie: Big Endian 2 - kolejność słów: Big Endian; kolejność bajtów w słowie: Little Endian 3 - kolejność słów: Little Endian; kolejność bajtów w słowie: Little Endian
`BitFieldWidth`	Liczba bitów pola bitowego. Suma BitFieldWidth i Position powinna być <= DataWidth; 0 oznacza brak pola bitowego (pełna szerokość danych = DataWidth)
`BitFieldPosition`	Pozycja najmłodszego bitu pola bitowego. Suma BitFieldWidth i Position powinna być <= DataWidth.
`Divisor`	Dzielnik (skala) wartości
`InitialValueAccess`	Metoda pierwszego dostępu do wartości Value: 0 - wartość początkowa Value jest odczytywana z urządzenia 1 - wartość początkowa Value jest zapisywana do urządzenia
`Value`	Zwraca ostatnio odczytaną wartość oraz określa wartość początkową

Nazwa	Opis
`RawValue`	Nieprzeskalowana wartość rejestru
`IsValueValid`	Określa, czy wartość jest zgodna ze stanem obiektu
`ErrorCode`	Kod błędu: 1 – niedozwolona funkcja 2 – niedozwolony numer rejestru 3 – niedozwolona wartość danej 4 – uszkodzenie w przyłączonym urządzeniu 5 – potwierdzenie pozytywne 6 – brak gotowości, komunikat usunięty 7 – potwierdzenie negatywne 8 – błąd parzystości pamięci 0 - poprawny odczyt/zapis rejestru -2 - przekroczenie czasu odpowiedzi -3 - błąd ramki (problem ze zdekodowaniem odpowiedzi) -4 - nieoczekiwany rozmiar odpowiedzi -5 - nieoczekiwany kod odpowiedzi

METODY

Nazwa	Opis
`SetTransmissionSpeed`	Ustawia prędkość transmisji
`SetParity`	Ustawia sprawdzanie parzystości
`SetStopBits`	Ustawia liczbę bitów stopu
`SetDeviceAddress`	Ustawia adres urządzenia Modbus typu Slave
`SetResponseTimeout`	Ustawia czas przekroczenia odpowiedzi w krokach 25ms
`SetRefreshPeriod`	Ustawia okres odświeżania w krokach 5ms - wartość 0 wyłącza automatyczne odświeżanie
`SetRegisterAddress`	Ustawia adres obsługiwanego rejestru
`SetRegisterType`	Ustawia typ rejestru Modbus
`SetInputOutputCount`	Ustawia liczbę dyskretnych wejść / wyjść podlegających operacji odczytu / zapisu
`SetDataType`	Ustawia typ zmiennej
`SetDataWidth`	Ustawia szerokość danych
`SetEndianness`	Ustawia typ kolejności bajtów
`SetBitFieldWidth`	Ustawia liczbę bitów pola - 0 oznacza brak pola bitowego (pełna szerokość danych DataWidth)
`SetBitFieldPosition`	Ustawia pozycję najmłodszego bitu pola
`SetDivisor`	Ustawia dzielnik
`ReadValue`	Rozpoczyna odczyt wartości z urządzenia - czeka na zakończenie odczytu w przypadku braku OnValueRead
`WriteValue`	Zapisuje nową wartość do urządzenia

ZDARZENIA

Nazwa	Opis
`OnValueChange`	Zdarzenie wywoływane w przypadku zmiany wartości cechy Value lub zmiany parametrów obiektu
`OnValueRead`	Zdarzenie wywoływane po zakończeniu odczytu rozpoczętego przez metodę ReadValue
`OnError`	Zdarzenie wywoływane gdy urządzenie slave zgłasza błąd

XV. Moduł GATE HTTP

> **Uwaga!** Opisana funkcjonalność oraz integracja jest dostępna dla **GRENTON GATE HTTP, DIN, Eth (INT-211-E-01)** posiadający **firmware 1.1.0-2034C lub wyższy!**

1. Informacje ogólne

Moduł GATE HTTP to urządzenie umożliwiające systemową integrację z zewnętrznymi serwisami posługującymi się protokołem HTTP, a także szeroką grupą urządzeń i systemów zewnętrznych/trzecich firm - np. urządzeń AV z interfejsami HTTP.

> **Uwaga!** Dla tworzonych obiektów wirtualnych nie ma ograniczenia względem ilości obiektów - ograniczeniem jest pamięć urządzenia, na którą wpływa m.in. poziom rozbudowania logiki na module.

2. Konfiguracja modułu

> **Uwaga!** Przed rozpoczęciem jakiejkolwiek pracy z modułem GATE HTTP wymagana jest aktualizacja bazy interfejsów!

Ustawianie czasu za pomocą serwera NTP

Moduł GATE HTTP umożliwia ustawianie czasu za pomocą serwera NTP wraz z uwzględnieniem strefy czasowej a także zmianą czasu (zimowy/letni). Czas pobierany jest automatycznie z serwera NTP (*pool.ntp.org*).

Do konfiguracji służą trzy cechy:

- `UseNTP` – określa czy GATE używa NTP,
- `NTPTimeout` - czas oczekiwania na odpowiedź z serwera NTP,
- `TimeZone` – ustawianie strefy czasowej GATE – dostępne są 22 strefy.

> **Uwaga!** Pobieranie czasu z serwera NTP wymaga, aby GATE znajdowało się w sieci, która posiada połączenie z internetem.

2.1. Obiekty wirtualne

2.1.1. HTTPRequest

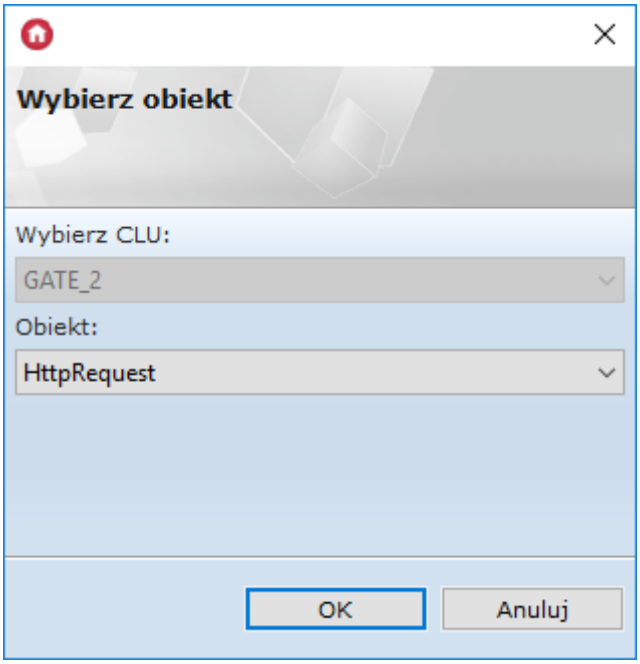
Dla HttpRequest przykładowo wykorzystywany jest serwis pogodowy http://api.openweathermap.org

Według przykładu na stronie openweathermap.org, zapytanie API wygląda następująco:

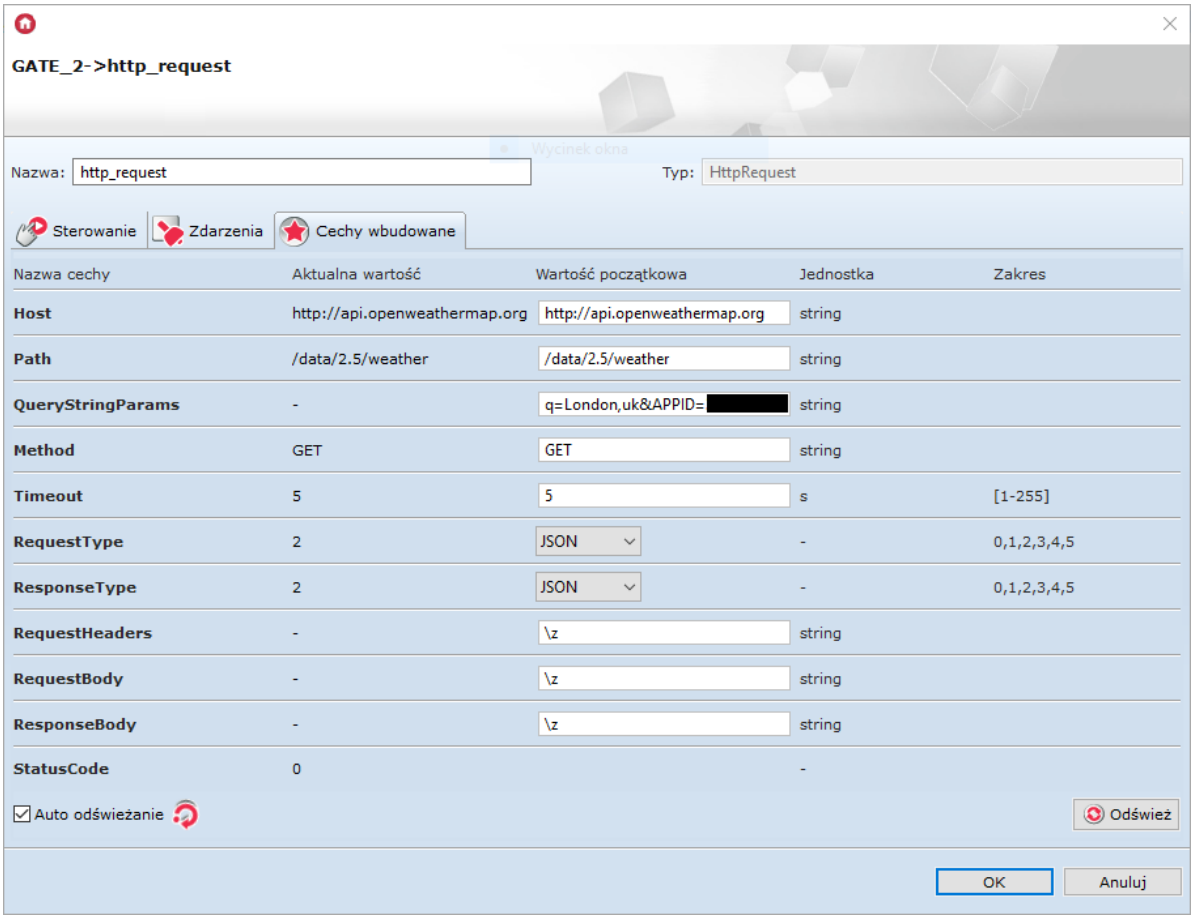
API call: http://api.openweathermap.org/data/2.5/weather?q=London&APPID={APIKEY}

HttpRequest - służy do wysyłania zapytań HTTP (typu GET, POST) do określonego hosta. Obsługiwane są standardowe Typy zawartości (content-type) m.in. JSON, XML.

Aby zastosować moduł Gate do odbierania zapytań, należy utworzyć obiekt wirtualny HttpRequest:



- W obiekcie HttpRequest należy ustawić następujące parametry:



- **Host:** api.openweathermap.org
- **Path:** /data/2.5/weather
- **QueryStringParams:** q=London&APPID={APIKEY}
- **Method:** GET
- **RequestType:** JSON

- **ResponseType:** JSON

> **Uwaga!** Obiekt Gate Http umożliwia obsługę połączeń szyfrowanych TLS. Jeżeli wymagane jest takie połączenie, należy na początku wartości w polu Host podać 'https://' . Jeżeli wartość nie zostanie podana, zostanie wykorzystane standardowe połączenie http.

> **Uwaga!** Gate Http nie obsługuje wszystkich połączeń szyfrowanych TLS, dlatego zalecamy przetestowanie połączenia z danym hostem.

> **Uwaga!** Podczas połączenia https czas nawiązania połączenia oraz otrzymania odpowiedzi od hosta jest dłuższy niż w przypadku połączenia http, dlatego należy zwiększyć wartość dla parametru Timeout.

> **Uwaga!** Cechy opisane jako nieustawialne są cechami zawierającymi odpowiedzi. Wartości początkowe tych cech należy pozostawić niezmienione. Wszelkie operacje na tych zmiennych należy wykonywać na skryptach (oraz zmiennych lokalnych).

Po wysłaniu konfiguracji i wywołaniu Metody SendRequest, StatusCode przyjmuje wartość 200 (OK).

Otrzymana odpowiedź na zapytanie jest przetrzymywana w cesze ResponseBody. Dla ustawionego ResponseType JSON, odpowiedź jest parsowana z json do tabeli. Wartość cechy jest niewidoczna z poziomu OM. Wartości odpowiedzi należy wyciągnąć z odpowiedzi z poziomu skryptu.

2.1.2. Pobieranie określonych wartości z otrzymanej odpowiedzi (XML,JSON)

> **Uwaga!** Uzyskaną odpowiedź ResponseBody należy przypisać do zmiennej lokalnej (w skrypcie).
>
> Przykładowo:
>
> local resp = GATE->httpr_openweather_json->ResponseBody
>
> Następnie w skryptach należy wykonywać operację na zmiennej resp!

> **Uwaga!** Skrypty odczytujące zawartość przetrzymywaną w cesze ResponseBody muszą być wykonane na module GATE HTTP.

Otrzymane odpowiedzi w zależności od ich typu (ResponseType) są odpowiednio parsowane do postaci tabeli.

Przykładowe odczyty wartości są zapisywane do zmiennych lokalnych (wewnątrz skryptu).

Aby była możliwość wykorzystania zmiennej poza skryptem (np. do wyświetlania w aplikacji), należy ją przypisać do zmiennych globalnych (cech użytkownika).

Poniżej przykładowe odpowiedzi w formacie XML oraz JSON oraz sposób odczytania danej wartości (w przedstawionych przykładach wykorzystano odpowiedzi z serwisu pogodowego openweathermap.org)

A. JSON:

Przykładowa odpowiedź (openweathermap.org):

```
resp = [[
{"coord":
{"lon":145.77,"lat":-16.92},
"weather":[{"id":803,"main":"Clouds","description":"broken
clouds","icon":"04n"}],
"base":"cmc stations",
"main":
{"temp":293.25,"pressure":1019,"humidity":83,"temp_min":289.82,"temp_max":295.3
7},
"wind":{"speed":5.1,"deg":150},
"clouds":{"all":75},
"rain":{"3h":3},
"dt":1435658272,
"sys":
{"type":1,"id":8166,"message":0.0166,"country":"AU","sunrise":1435610796,"sunse
t":1435650870},
"id":2172797,
"name":"Cairns",
"cod":200}
]]
```

Jak odczytać:

- Wartość parametru **lon**

```
{"coord":
{"lon":145.77,"lat":-16.92},
"weather":[{"id":803,"main":"Clouds","description":"broken
clouds","icon":"04n"}],
"base":"cmc stations",
"main":
{"temp":293.25,"pressure":1019,"humidity":83,"temp_min":289.82,"temp_max":295.3
7},
```

W skrypcie:

```
    local lon = resp.coord.lon
```

Po wywołaniu skryptu do zmiennej lokalnej (zmienna skryptu) zostanie przypisana wartość *145.77* .

- Wartość parametru **description**

```
{"coord":
{"lon":145.77,"lat":-16.92},
"weather":[{"id":803,"main":"Clouds","description":"broken
clouds","icon":"04n"}],
"base":"cmc stations",
"main":
{"temp":293.25,"pressure":1019,"humidity":83,"temp_min":289.82,"temp_max":295.3
7},
```

W skrypcie:

```
local description = resp.weather[1].description
```

Po wywołaniu skryptu do zmiennej lokalnej (zmienna skryptu) zostanie przypisana wartość *„broken clouds”.*

B. XML:

Przykładowa odpowiedź (openweathermap):

```
resp= [[
<current>
   <city id="2643741" name="City of London">
    <coord lon="-0.09" lat="51.51">
    <country>GB</country>
    <sun rise="2015-06-30T03:46:57" set="2015-06-30T20:21:12">
   </city>
   <temperature value="72.34" min="66.2" max="79.88" unit="fahrenheit"/>
   <humidity value="43" unit="%">
   <pressure value="1020" unit="hPa">
   <wind>
    <speed value="7.78" name="Moderate breeze">
    <direction value="140" code="SE" name="SouthEast">
   </wind>
   <clouds value="0" name="clear sky">
   <visibility value="10000">
   <precipitation mode="no">
   <weather number="800" value="Sky is Clear" icon="01d">
   <lastupdate value="2015-06-30T08:36:14">
</current>
]]
```

Jak odczytać:

- Wartość atrybutu id w tagu **city**

```
<current>
    <city id="2643741" name="City of London">
    <coord lon="-0.09" lat="51.51">
    <country>GB</country>
<sun rise="2015-06-30T03:46:57" set="2015-06-30T20:21:12">
</city>
```

W skrypcie:

```
local  city_id = resp[1].id
```

Po wywołaniu skryptu do zmiennej lokalnej (zmienna skryptu) zostanie przypisana wartość *2643741*.

- Wartość znajdująca się pomiędzy tagiem **country**

```
 <current>
    <city id="2643741" name="City of London">
     <coord lon="-0.09" lat="51.51">
     <country>GB</country>
     <sun rise="2015-06-30T03:46:57" set="2015-06-30T20:21:12">
   </city>
```

W skrypcie:

```
local  country = resp[1][2][1]
```

Po wywołaniu skryptu do zmiennej lokalnej (zmienna skryptu) zostanie przypisana wartość *„GB”*.

- Nazwa tagu **country**

```
<current>
 <city id="2643741" name="City of London">
    <coord lon="-0.09" lat="51.51">
    <country>GB</country>
    <sun rise="2015-06-30T03:46:57" set="2015-06-30T20:21:12">
  </city>
```

W skrypcie:

```
local nameTag = resp[1][2].xmlTag
```

Po wywołaniu skryptu do zmiennej lokalnej (zmienna skryptu) zostanie przypisana wartość „country”.

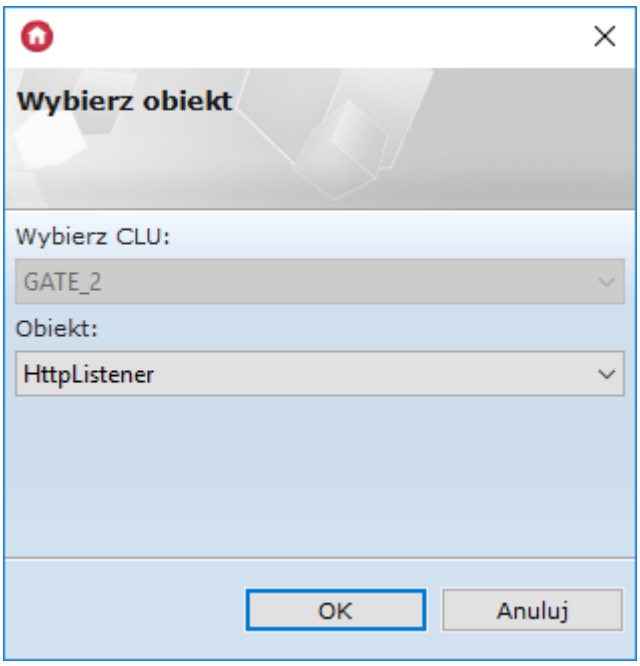
2.2.1. HttpListener

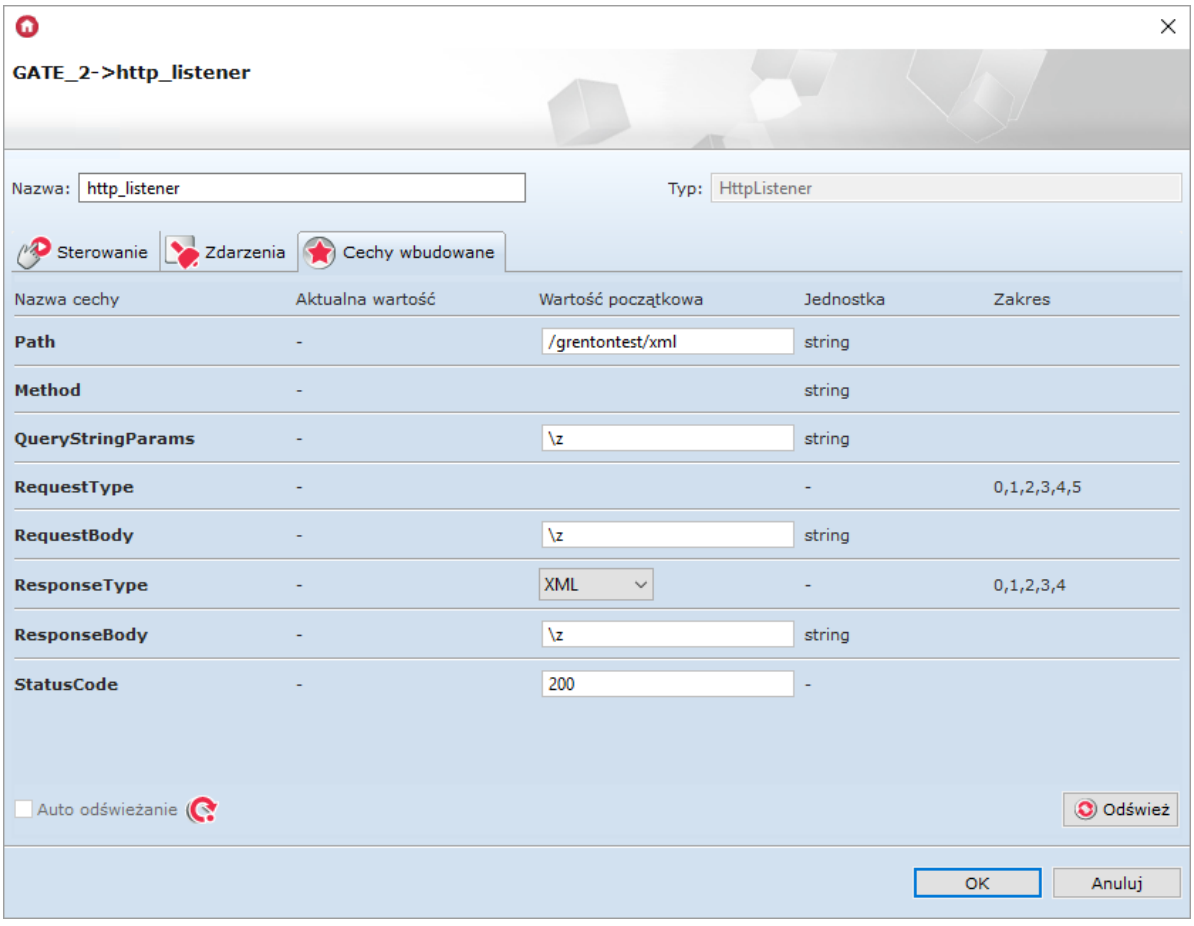
Obiekt HttpListener służy do otrzymywania zapytań HTTP (typu GET, POST). Wysyłana odpowiedź zwrotna może być serializowana do jednego ze standardowych typów m.in. JSON, XML. W obiekcie HttpListener ważne jest, aby na każdy przychodzący Request odesłać odpowiedź (Response).

W przypadku nasłuchiwania na zapytanie Request do modułu Gate - przykładowo (korzystając np. z przeglądarki internetowej):

GET 192.168.4.12/grentontest/xml

Należy utworzyć obiekt wirtualny HttpListener





W obiekcie HttpListener należy ustawić następujące parametry:

- **Path:** /grentontest/xml
- **ResponseType:** XML
- **StatusCode:** 200

> **Uwaga!** Cechy opisane jako nieustawialne są cechami zawierającymi odpowiedzi. Wartości początkowe tych cech należy pozostawić niezmienione. Wszelkie operacje na tych zmiennych należy wykonywać na skryptach (oraz zmiennych lokalnych)

Do zdarzenia OnRequest należy utworzyć skrypt, który będzie tworzył poprawną odpowiedź i wysyłał ją zwrotnie.

2.2.2. Przygotowanie odpowiedzi wysyłanej do serwera

Odpowiedź jest tworzona w zmiennej lokalne resp.

Po przygotowaniu odpowiedzi należy ją ustawić dla cechy ResponseBody(resp), a następnie wysłać za pomocą metody SendResponse()

A. XML:

Aby w odpowiedzi wysłać wartość danej cechy:

```
local resp ="<clu><temperature>" ..CLUZ->x103478262_ONEW_SENSOR1->Value.."
</temperature></clu>"
GATE_2->Listener_XML->SetResponseBody(resp)
GATE_2->Listener_XML->SendResponse()
```

Przesłana odpowiedź wygląda następująco:

```
<clu>
  <temperature>22.5</temperature>
</clu>
```

B.JSON:

```
local resp = {
Temp = CLUZ->x103478262_ONEW_SENSOR1->Value
}
GATE_2->Listener_JSON->SetResponseBody(resp)
GATE_2-> Listener_JSON->SendResponse()
```

Przesłana odpowiedź wygląda następująco:

```
{"Temp":22.6}
```

2.2.3. Odczyt wartości kluczy z parametru querystringparams

Zgodnie z opisem cechy QueryStringParams jej wartość nie jest ustawialna, można odczytać ją w skrypcie. Jeżeli w zapytaniu zostaje wysłane querystring z kluczami (keys), to z poziomu skryptu można odczytać daną wartość - jest zapisana w postaci tabeli.

Poszczególne wartości kluczy można uzyskać na zasadzie:

```
value1 = qs.klucz1
```

Dla otrzymanego zapytania:

192.168.1.12/grentontest/query?light1=on&light2=off&light3=on

Należy utworzyć skrypt:

```
local qs = HTTP_L->grentontest_query_listener->QueryStringParams

local test0 = qs.light1
local test1 = qs.light2
local test2 = qs.light3

HTTP_L->grentontest_query_listener->SetResponseBody()
HTTP_L->grentontest_query_listener->SendResponse()
```

Wszystkie wartości kluczy zostaną zapisane w zmiennych lokalnych (test0, test1, test2).

2.3.1. Timer

Timery są wirtualnymi obiektami tworzonymi w ramach danego modułu GATE. Timery mogą być wykorzystywane wszędzie tam, gdzie potrzebne jest wywołanie metody po określonym czasie lub też jej cykliczne wywoływanie.

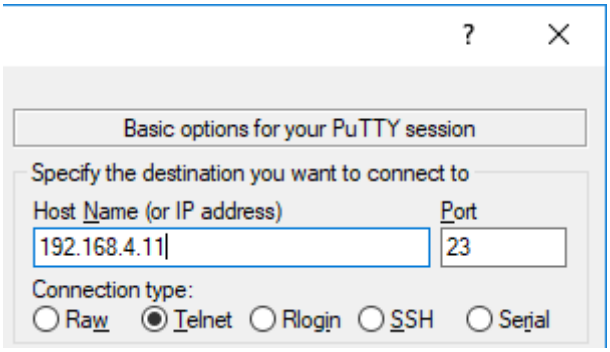
> **Uwaga!**
>
> Zalecane jest wykorzystywanie obiektu Timer w przypadku cyklicznego wysyłania zapytań za pomocą obiektu HttpRequest.

Timer może pracować w dwóch trybach:

- `Countdown` Po wystartowaniu, odlicza ustalony czas. Po zakończeniu odliczania uruchamiana jest metoda powiązana ze zdarzeniem `OnTimer`, a timer zatrzymuje się i nie odlicza, aż do następnego uruchomienia metodą `Start`.
- `Interval` Timer cykliczny - po starcie zaczyna odliczać ustawiony czas. Po jego upłynięciu timer wywołuje metodę powiązaną ze zdarzeniem `OnTimer`, a sam timer ponownie zaczyna odliczać zadany czas. Sytuacja powtarza się, aż do momentu zatrzymania metodą `Stop`.

3. Możliwość połączenia z Gate za pomocą TELNET

Dla modułu Gate Http możliwy jest podgląd skryptów Lua. W przypadku błędu konfiguracji (tryb emergency) możliwy jest podgląd miejsca błędu w utworzonej konfiguracji LUA. Połączenie jest nawiązywane za pomocą protokołu Telnet - w tym celu można wykorzystać np. program PuTTY. Przykładowe parametry do nawiązania połączenia:



Do wywołania połączenia po stronie Gate'a wykorzystane mogą być dwie metody:

- `StartConsole` - Uruchamia konsolę Lua. W momencie wywołania metody, użytkownik ma 10s na ustawienie połączenia z Gate. Przy poprawnym połączeniu, na terminalu (klient) zostanie zwrócona

informacja o poprawnym połączeniu:

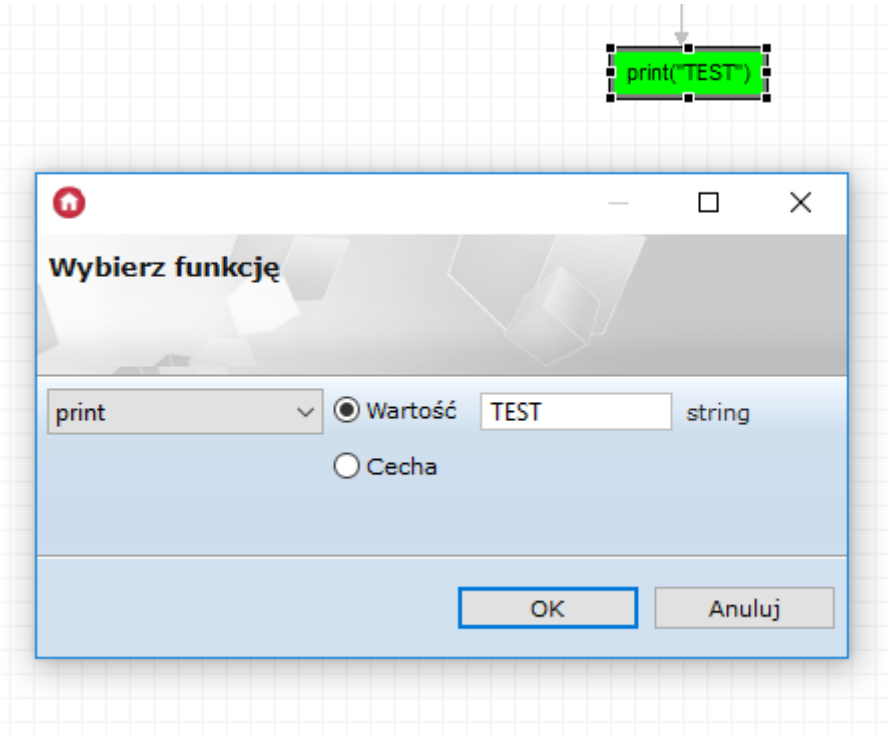
```
CLU SN Telnet session started.
```

- `StartConsoleOnReboot` – umożliwia nawiązanie połączenia przy następnym restarcie Gate'a. Po restarcie użytkownik ma 10s na ustawienie połączenia z Gate. Przy poprawnym połączeniu, na terminalu (klient) zostanie zwrócona informacja o poprawnym połączeniu

```
CLU SN initializing...
CLU: running user.lua...
CLU: running om.lua...
CLU: running OnInit...
CLU: Project loaded.
```

Uwaga! Niezalecane jest przypisanie metody `StartConsole` oraz `StartConsoleOnReboot` do zdarzenia `OnInit` modułu GATE Http.

Aby na konsoli wyświetlić np. wartość danej cechy, należy wykorzystać komponent *Blok funkcyjny* i wybrać metodę *Print*, a następnie wybrać żądaną cechę.



Uwaga! Jeżeli połączenie Telnet pomiędzy klient <--> GATE zostało nawiązane, ponowne wywołanie metody StartConsole spowoduje rozłączenie bieżącej sesji.

4. Kompleksowa integracja z systemami zewnętrznymi przy użyciu urządzenia GATE HTTP

Opis konfiguracji krok po kroku na przykładzie wyjścia przekaźnikowego.

4.1. System

Powiedzmy, że mamy prosty system złożony z następujących elementów:

- CLU Z-Wave - nazwany (Name) "CluZ"
- Moduł przekaźnikowy - na potrzeby przykładu użyjemy jednego wyjścia o nazwie "Relay"
- Gate Http - nazwa "GateHttp"

4.2. Sterowanie wyjściem

W celu umożliwienia sterowania wyjściem przekaźnikowym z zewnętrznego systemu tworzymy nowy obiekt typu HttpListener na GateHttp i konfigurujemy jak poniżej:

- **Name:** RelayControlListener
- **Path:** /relaycontrol

Pozostałe parametry pozostawiamy na razie bez zmian.

Skrypt

Aby obiekt `RelayControlListener` zadziałał należy utworzyć dla skrypt, który będzie obsługiwał przychodzące zapytania Http.

Tutaj warto zauważyć, że z tego skryptu mamy dostęp do całego systemu i wszystkich jego funkcjonalności. To otwiera praktycznie nieograniczone możliwości, ale też rodzi pewne ryzyka, zwłaszcza jeśli funkcjonalność Gate'a nie jest dobrze przemyślana. Dlatego zwracamy szczególną uwagę, że implementując funkcjonalność Gate'a należy dobrze przemyśleć sposób działania jaki chcemy osiągnąć oraz to jak działanie Gate'a może zależeć albo wpływać na inne elementy systemu. Przykłady takiego podejścia będą też omawiane dalej.

Wracając do skryptu kontrolującego Relay. Chcemy mieć możliwość załączenia lub rozłączenia Relay'a wysyłając do niego oczekiwany stan (On/Off) albo wykonania metody `Switch`. Takie podejście do implementacji umożliwia podłączenie do niego zarówno kontroli typu przełącznika bistabilnego jak i monostabilnego.

Przechodząc do działania, tworzymy na GateHttp skrypt o nazwie `RelayControlOnRequest`, i w trybie edycji kodu wrzucamy to co poniżej:

```
-- RelayControlOnRequest()
local data = GateHttp->RelayControlListener->QueryStringParams
if data == nil then
    CluZ->Relay->Switch(0)
else
    if data.cmd == "setValue" then
        local val = tonumber(data.val)
        if(val == 1) then
            CluZ->Relay->SwitchOn(0)
        elseif(val == 0) then
            CluZ->Relay->SwitchOff(0)
        end
    end
end

GateHttp->RelayControlListener->StatusCode = 200
GateHttp->RelayControlListener->ResponseBody = "OK"
GateHttp->RelayControlListener->SendResponse()
```

Następnie przypisujemy skrypt do zdarzenia `OnRequest` obiektu `RelayControlListener` i wysyłamy konfigurację do systemu.

Powyższy skrypt pobiera z obiektu `RelayControlListener` wartości parametrów zapytania i zależnie od tego co w nich się znajduje wykonuje odpowiednie akcje. Po czym odsyła do klienta status operacji - w tym przypadku `200, OK`.

Działanie można łatwo przetestować za pomocą zwykłej przeglądarki internetowej wpisując poniższe URL'e (oczywiście adres IP należy zamienić na rzeczywisty adres Waszego Gate'a Http):

`http://192.168.88.4/relaycontrol?cmd=setValue&val=1` - Włącza Relay

`http://192.168.88.4/relaycontrol?cmd=setValue&val=0` - Wyłącza Relay

`http://192.168.88.4/relaycontrol` - Przełącza stan Relay'a

Jak widać na przykładach możemy użyć Listenera na dwa sposoby. Jeśli parametr `cmd` (komenda do wykonania) i `val` (wartość do ustawienia) są odpowiednio zdefiniowane to ustawiają konkretny stan Relay'a. Jeśli pominiemy w URL'u te parametry to obiekt działa jak przełącznik (Switch).

Powyższy przykład może zostać dalej rozbudowany o kolejne komendy, jeśli jest potrzeba wykonania innych komend. Można też dodać kolejne parametry identyfikujące obiekt, na którym te komendy należy wykonać.

4.3. Pobieranie stanu

W poprzednim kroku umożliwiliśmy sterowania obiektem w systemie z zewnątrz. Bardzo często w kolejnym kroku pojawia się potrzeba udostępnienia także możliwości pobrania aktualnego stanu obiektu.

Jedną z szybszych i najbardziej intuicyjnych metod (niekoniecznie najlepszą) jest zdefiniowane kolejnego Listenera który pobierze wartość `Value` z obiektu Relay i odeśle do klienta. Najprostszy skrypt realizujący taką funkcjonalność wyglądać może jak poniżej.

```
-- RelayStateOnRequest()
GateHttp->RelayState->StatusCode = 200
GateHttp->RelayState->ResponseBody = "Relay State: "..CluZ->Relay->Value
GateHttp->RelayState->SendResponse()
```

Wpisując poniższy URL do przeglądarki widać, że dostajemy odpowiedź ze stanem obiektu Relay (w prostej postaci tekstowej, ale to nie format przysłania danych jest tematem tego przykładu).

`http://192.168.88.4/relaystate` - zwraca `Relay State: 0` lub `Relay State: 1` zależnie od stanu obiektu.

Powyższy przykład na pierwszy rzut oka działa dobrze, ale spróbujmy się przyglądnąć bliżej.

4.4. Kolejność zdarzeń

Skonstruowaliśmy właśnie interfejs (API) Http posiadający dwie metody:

- **/relaycontrol** - umożliwia kontrolowanie obiektem Relay
- **/relaystate** - zwraca aktualny stan (wartość) obiektu Relay

Po szybkim przetestowaniu wszystko działa dobrze, ale jak pisaliśmy wyżej, należy jeszcze zastanowić się jak takie metody zostaną użyte. Mianowicie, łatwo sobie wyobrazić, że w zewnętrznym systemie te dwie metody zostaną użyte tuż po sobie: wywołanie akcji przełączenia oraz po otrzymaniu odpowiedzi, odczytanie stanu w celu potwierdzenia, że akcja nastąpiła i synchronizacji statusu.

I tutaj może nastąpić nieoczekiwane działanie sytemu - Relay się załącza, ale zwracany stan jest 0, czyli nieprawidłowy. Przyczyną takiego stanu rzeczy jest fakt, że operacje te są wykonywane asynchronicznie na dwóch różnych urządzeniach. Nie ma gwarancji, że operacja zmiany stanu Relay'a zdąży się wykonać zanim zapytamy o jego stan. Wywołanie akcji zmiany stanu w skrypcie `RelayControlOnRequest()` wywoływane jest asynchronicznie, co oznacza, że skrypt nie czeka aż CluZ wykona zadanie.

Rozważany przypadek jest bardzo prosty i praktycznie zawsze zadziała, ale w przypadku bardziej skomplikowanych operacji (gdy zaangażowane są różne obiekty celu, do wykonania operacji konieczna jest jeszcze wymiana danych, przesłanie cech, itd.) ryzyko, że pobranie statusu nastąpi przed realną zmianą stanu obiektu(ów) jest realny i w złożonych systemach często obserwujemy takie efekty.

4.5. Synchronizacja zdarzeń

Powyższy problem można rozwiązać zmuszając skrypt `RelayControlOnRequest()` aby zaczekał, aż CluZ realnie wywoła na docelowym urządzeniu akcję zmiany stanu. Można to w prosty sposób osiągnąć za pomocą funkcji `clu.await()`. Np. wywołanie:

```
CluZ->Relay->Switch(0)
```

zamieniamy na:

```
clu.await(CluZ->Relay->Switch(0))
```

(Pozostałe wywołania akcji na CluZ zamieniamy w analogiczny sposób).

Od teraz nasz Listener nie wyśle potwierdzenia `200, OK` zanim akcja na CluZ nie zostanie realnie wykonana więc klient korzystający z tego interfejsu nie zostanie wprowadzony w błąd poprzez zbyt szybkie potwierdzenie wykonania zadania.

Funkcja `clu.await()` ma jednak pewne ograniczenie. Limit czasowy na wykonanie wywołania jest 800ms i jeśli nie uda się zrealizować zadania w takim czasie, skrypt zakończy się timeout'em i klient Http dostanie w odpowiedzi błąd Http: `500 Internal Server Error`.

W większości przypadków takie timeout nie jest problemem i system będzie działał poprawnie, ale w przypadku złożonych operacji i/lub gdy CluZ będzie obciążone innymi zadaniami może się to wydarzyć. Sposób na rozwiązanie problemu w takim przypadku został opisany w kolejnej sekcji.

4.6. Potwierdzenie zwrotne

W złożonych systemach oraz tam, gdzie zależy nam na dużej niezawodności i stabilności działania integracji należy opóźnić odpowiedź Listenera Http do czasu otrzymania wprost potwierdzenia z CluZ, że zadanie zrealizowano.

W tym celu rozbijamy działanie na dwa etapy. Zamiast jednego skryptu realizującego całość zadania definiujemy dwa: pierwszy realizuje zadanie, drugi wysyła odpowiedź Http po potwierdzeniu przez CluZ, że zakończono wykonywanie zadania.

Dla przejrzystości definiujemy nowy skrypt i przypisujemy go do Listenera:

- Event `OnRequest`: GateHttp->SplitSyncOnRequest()

Skrypt `SplitSyncOnRequest()` wygląda następująco:

```
-- SplitSyncOnRequest()
local data = GateHttp->RelayControlListener->QueryStringParams
if data == nil then
    CluZ->SplitSyncCluzTask("Switch")
    return
else
    if data.cmd == "setValue" then
        local val = tonumber(data.val)
        if(val == 1) then
            CluZ->SplitSyncCluzTask("On")
            return
        elseif(val == 0) then
            CluZ->SplitSyncCluzTask("Off")
            return
        end
    end
end

GateHttp->RelayControlListener->StatusCode = 400
GateHttp->RelayControlListener->ResponseBody = "Bad request"
GateHttp->RelayControlListener->SendResponse()
```

W każdym miejscu skryptu, gdy delegujemy zadnie do CluZ (tym razem poprzez dodatkowy skrypt o czym za chwilę), kończymy działanie naszego skryptu bez wysyłania odpowiedzi Http do klienta. Jeśli wykonanie skryptu dojdzie do końcowych linii będzie to oznaczać, że nie udało się prawidłowo zinterpretować zapytania co oznacza, że jest ono nieprawidłowo i odsyłamy błąd `400, Bad request`. Przy okazji dodaliśmy kolejny poziom zabezpieczenia przed nieprawidłowymi parametrami wywołania.

Zadanie nie jest teraz jak poprzednio wykonywane bezpośrednio na docelowym obiekcie Relay, ale delegowane do skryptu na CluZ o nazwie `SplitSyncCluzTask(action: string)`. Zastosowana notacja oznacza, że skrypt jest wywoływany z parametrem o nazwie `action`, który jest typu `string` - nie mylić z notacją LUA gdzie nie definiujemy typu parametru wywołania funkcji. Parametr `action` definiuje konkretną akcję do wywołania na obiekcie Relay. Działanie jest identyczne jak w poprzednim przypadku.

```
-- SplitSyncCluzTask(action: string)
if(action == "On") then
    CluZ->Relay->SwitchOn(0)
elseif (action == "Off") then
    CluZ->Relay->SwitchOff(0)
```

```
elseif (action == "Switch") then
    CluZ->Relay->Switch(0)
else
    -- Unknown action
    GateHttp->SplitSyncRequestCompleted(false)
  -- Return to avoid double completion
  return
end
GateHttp->SplitSyncRequestCompleted(true)
```

Zależnie od zdefiniowanej akcji wykonywana jest odpowiednia metoda na obiekcie Relay. Na końcu informujemy GateHttp, że zakończyliśmy zadanie i należy odesłać do klienta odpowiedź. Została do tego celu stworzona metoda na GateHttp o nazwie `SplitSyncRequestCompleted(success: boolean)`, która jako parametr przyjmuje wartość logiczną: `true` jeśli udało się wykonać akcję, `false` w przeciwnym przypadku.

```
-- SplitSyncRequestCompleted(success: boolean)
if success then
    GateHttp->RelayControlListener->StatusCode = 200
    GateHttp->RelayControlListener->ResponseBody = "OK"
else
    GateHttp->RelayControlListener->StatusCode = 405
    GateHttp->RelayControlListener->ResponseBody = "Not allowed"
end
GateHttp->RelayControlListener->SendResponse()
```

GateHttp poprzez powyższą metodę wysyła odpowiedź do klienta informującą o sukcesie lub błędzie zależnie od otrzymanego parametru. Tym sposobem zrealizowaliśmy w pełni synchroniczną metodę Http, która nie ma ograniczenia czasowego na działanie. W bardziej zaawansowanych przypadkach można jeszcze bardziej usprawnić działanie systemu poprzez wywołanie funkcji `SplitSyncRequestCompleted(success: boolean)` w odpowiedzi na zdarzenia informujące o zmianie wartości konkretnego obiektu. Co daje pewność, że zmiana nastąpiła i jeszcze bardziej zwiększa stabilność systemu.

> **Uwaga!** W przypadku danych otrzymywanych z zewnętrznych systemów należy zawsze stosować metodę ograniczonego zaufania co do ich poprawności. Zalecamy nieprzekazywanie bezpośrednio wartości do metod i skryptów wewnątrz systemu a stosowanie konkretnych akcji w zależności od wartości metod jak widać w powyższych skryptach. W przypadku konieczności bezpośredniego użycia zmiennych otrzymanych z zewnątrz należy je przekazywać za pośrednictwem zmiennych użytkownika (które są adresowalne w całym systemie Grenton i można je swobodnie przesyłać między urządzeniami CLU). Dodatkowo każdą zmienną otrzymaną z zewnątrz należy w skrypcie walidować pod kątem jest poprawności, wartości, zakresu. Brak odpowiedniej weryfikacji otrzymanych wartości może powodować nieoczekiwane działanie systemu, otworzyć dostęp do niechcianych funkcjonalności a nawet powodować błędy oraz przejście CLU w tryb Emergency.

4.7. Timeout

Stworzony Listener działa już prawie niezawodnie. Dlaczego prawie? Zastanówmy się co się stanie, jeśli CluZ z jakiegoś powodu nigdy nie wywoła metody `SplitSyncRequestCompleted(success: boolean)`. GateHttp zostaje wtedy w stanie oczekiwania na zakończenie obsługi bieżącego zapytania i przestaje reagować na kolejne zapytania.

Oczywiście w dobrze skonfigurowanym systemie nie powinno się to zdarzyć. Jednak zawsze może wystąpić niespodziewana sytuacja i dlatego każdy element systemu powinien być skonfigurowany tak aby działał możliwie niezależnie i był odporny na błędy w innych obszarach. Dlatego nasz Listener też powinien być w pełni odporny na takie sytuacje.

W tym celu na GateHttp zdefiniujemy obiekt Timer, który będzie pilnował, żeby oczekiwanie na odpowiedź CluZ nie trwała w nieskończoność. Parametry nowego obiektu:

- **Name:** SplitSyncTimeout
- **Event OnTimer:** GateHttp->SplitSyncTimeoutOnTimer()
- **Time:** 3000 - tutaj należy dobrać czas odpowiednio do konkretnej sytuacji, na potrzeby przykładu przyjmujemy 3s (3000ms)
- **Mode:** CountDown

Skrypt wykonywany po upłynięciu zadanego czasu wygląda następująco:

```
-- SplitSyncTimeoutOnTimer()
GateHttp->RelayControlListener->StatusCode = 408
GateHttp->RelayControlListener->ResponseBody = "Timeout"
GateHttp->RelayControlListener->SendResponse()
```

Działanie skryptu jest dość proste, po prostu zwraca błąd `408, Timeout`.

Żeby wszystko zadziałało należy odpowiednio zmodyfikować skrypty `SplitSyncOnRequest()` oraz `SplitSyncRequestCompleted(success: boolean)`.

```
-- SplitSyncOnRequest()
local data = GateHttp->RelayControlListener->QueryStringParams
if data == nil then
    CluZ->SplitSyncCluzTask("Switch")
    GateHttp->SplitSyncTimeout->Start()
    return
else
    if data.cmd == "setValue" then
        local val = tonumber(data.val)
        if(val == 1) then
            CluZ->SplitSyncCluzTask("On")
            GateHttp->SplitSyncTimeout->Start()
            return
        elseif(val == 0) then
            CluZ->SplitSyncCluzTask("Off")
            GateHttp->SplitSyncTimeout->Start()
            return
        end
    end
end

GateHttp->RelayControlListener->StatusCode = 400
```

```
GateHttp->RelayControlListener->ResponseBody = "Bad request"
GateHttp->RelayControlListener->SendResponse()
```

Za każdym razem, gdy delegujemy zadanie do CluZ startujemy Timer `SplitSyncTimeout`.

```
-- SplitSyncRequestCompleted(success: boolean)
if(GateHttp->SplitSyncTimeout->State == 1) then
    GateHttp->SplitSyncTimeout->Stop()
    if success then
        GateHttp->RelayControlListener->StatusCode = 200
        GateHttp->RelayControlListener->ResponseBody = "OK"
    else
        GateHttp->RelayControlListener->StatusCode = 405
        GateHttp->RelayControlListener->ResponseBody = "Not allowed"
    end
    GateHttp->RelayControlListener->SendResponse()
end
```

Natomiast w skrypcie `SplitSyncRequestCompleted(success: boolean)` sprawdzamy najpierw czy Timer jest ciągle w stanie `1` (włączonym). Zapobiega to przed niepotrzebną próbą wysyłania odpowiedzi w sytuacji, gdy timeout już wystąpił - czas na odpowiedź się skończyć i została wysłana odpowiedź informująca o wystąpieniu błędu `408, Timeout`. Jeśli Timer ciągle działa (normalna sytuacja, czas na odpowiedź się nie wyczerpał) zatrzymujemy Timer i dalej postępujemy tak jak poprzednio.

4.8. Wiele obiektów

Wróćmy jeszcze na chwilę do metody pobierającej stan Relay'a. W szczególności przyjrzyjmy się jeszcze raz następującej linii:

```
GateHttp->RelayState->ResponseBody = "Relay State: "..CluZ->Relay->Value
```

Kluczowe tutaj jest pobranie wartości cechy Value obiektu Relay:

```
CluZ->Relay->Value
```

Ta metoda działa dobrze, ale należy zdawać sobie sprawę, że wartość tej cechy pobierana jest w momencie wykonywania skryptu. W jej wyniku następuje komunikacje między GateHttp a CluZ przez sieć. Jest to wywołanie synchroniczne, czyli metoda czeka aż zostanie dostarczona odpowiedź z wartością cechy `Value` obiekty Relay. Wiemy już o pewnych ograniczeniach takiego wywołania. W tym konkretnym przypadku zagrożeń jest jeszcze więcej. Mianowicie, wartość tej cechy jest pobierana za każdym razem, gdy klient zapyta o jej wartość przez interfejs Http co generuje niepotrzebny ruch w systemie. Dodatkowo wprowadza niepotrzebne opóźnienie w systemie. Jeśli takich zapytań jest dużo to może to mieć wpływ na wydajność systemu. W niektórych zwłaszcza prostych przypadkach jest to akceptowalne i system będzie sobie z tym dobrze radził. Ale nie zawsze.

Wyobraźmy sobie, że w systemie jest wiele obiektów i potrzebujemy w odpowiedzi dostarczyć statusy wszystkich z nich (w postaci JSON lub CSV). Jeśli w takim przypadku zastosujemy identyczną metodę to skrypt realizujący takie zadanie może wyglądać mniej więcej jak poniżej:

```
```

```
GateHttp->RelayState->StatusCode = 200

local response = CluZ->Relay01->Value
response = response .. "," .. CluZ->Relay02->Value
response = response .. "," .. CluZ->Relay03->Value
response = response .. "," .. CluZ->Relay04->Value
response = response .. "," .. CluZ->Relay05->Value
response = response .. "," .. CluZ->Relay06->Value
response = response .. "," .. CluZ->Relay07->Value
response = response .. "," .. CluZ->Relay08->Value
response = response .. "," .. CluZ->Relay09->Value
response = response .. "," .. CluZ->Relay10->Value
response = response .. "," .. CluZ->Relay11->Value

GateHttp->RelayState->ResponseBody = "System State: ".. response
GateHttp->RelayState->SendResponse()
```

W rzeczywistym systemie obiektów Relay może być znacznie więcej. Każda linia powoduje odpytanie do CluZ o wartość cechy Value przez sieć. Zebranie stanu wszystkich obiektów może zabrać sporo czasu. Opóźnia to znacznie odpowiedź i na czas wykonywania operacji blokuje GateHttp.

Seria odpytań występuje każdorazowo, gdy klient zapyta o stan systemu. W większości przypadków wartość cechy między zapytaniami zmienia się tylko dla jednego obiektu, tego właśnie zmienionego. Te wszystko powoduje dużo niepotrzebnego ruchu i negatywnie wpływa na szybkość działania systemu. Z punktu widzenia użytkownika końcowego system może w takich przypadkach działać niestabilnie, mieć nieoczekiwane opóźnienia, zawieszać się na krótkie lub dłuższe chwile a nawet gubić niektóre zdarzenia.

4.9. Stan dla złożonego systemu

W celu rozwiązania powyższego problemu należy trochę inaczej podejść do zadania pobierania stanu urządzeń. W dalszej części tej sekcji, dla prostoty przykładów, wrócimy do pojedynczego obiektu Relay, ale podana metoda zadziała praktycznie dla dowolnej liczby obiektów.

Powiedzmy, że zamiast odpytywać zdalne CluZ o stan obiekty Relay za każdym razem, kiedy klient o niego zapyta moglibyśmy trzymać jego wartość lokalnie w zmiennej użytkownika GateHttp. Dzięki temu, kiedy klient zapyta bez żadnych opóźnień, zwracamy jej wartość natychmiast bez żadnych opóźnień, nazwijmy ją `RelayValueOnGateHttp`. Co więcej chcielibyśmy wyeliminować wszystkie zapytania synchronizujące jej wartość i dostawać informacje tylko wtedy, kiedy jest to potrzebne, czyli kiedy wartość cechy `CluZ->Relay->Value` się zmieni. Aby to osiągnąć do zdarzenia `OnValueChange` obiektu Relay przypisujemy następującą komendę:

```
GateHttp->RelayValueOnGateHttp=CluZ->Relay->Value
```

Co mniej więcej oznacza: Za każdym razem, kiedy wartość cechy Value się zmieni, przypisz do cechy użytkownika `RelayValueOnGateHttp` na GateHttp tą nową wartość. Od teraz po stronie GateHttp zawsze będziemy mieli aktualną wartość obiektu Relay. W momencie zapytania wysyłamy po prostu tą wartość do klienta. Aby to zrealizować modyfikujemy skrypt `RelayStateOnRequest()` w następujący sposób:

```
-- RelayStateOnRequest()
GateHttp->RelayState->StatusCode = 200
GateHttp->RelayState->ResponseBody = "Relay State: "..GateHttp-
>RelayValueOnGateHttp
GateHttp->RelayState->SendResponse()
```

Jak wspomniano wcześniej można zastosować dla dowolnie wielu obiektów i nie powoduje żadnego negatywnego wpływu na wydajność systemu, ponieważ komunikowane są tylko zmiany poszczególnych wartości w momencie, kiedy wystąpią.

4.10. Push Notyfikacje

Idąc krok dalej na drodze do idealnej integracji zaimplementujmy jeszcze jedno usprawnienie. Jak dotąd klient sam musiał dopytywać co chwilę czy przypadkiem coś nie zmieniło się w systemie. Jeśli system ma być responsywny to takie zapytania muszą odbywać się często. Częste zapytania generują niepotrzebny ruch i zwiększają ryzyko opóźnień, zwłaszcza w obsłudze zdarzeń bardzo wrażliwych na opóźnienia jak np.: włączanie oświetlenia, gdzie użytkownik od razu czuje, że akcja nie nastąpiła natychmiast po dotknięciu przycisku.

Dodatkowo klient nie jest notyfikowany natychmiast a zmianie w systemie, ale dopiero w momencie, kiedy sam dopyta czy aby nic się nie zmieniło.

Rozwiązaniem jest metoda Push stanu, gdzie to system sam aktywnie wysyła notyfikację a zmianie stanu urządzenia w systemie. W celu zaimplementowania takiego mechanizmu tworzymy nowy obiekt na GateHttp typu HttpRequest:

- **Name:** StatePushNotification
- **Host: IP:** Port serwera http nasłuchującego informacji o zmianach stanu
- **Path:** /statechanged
- **Method:** PUT

Pozostałe ustawienia bez zmian.

Następnie dodajemy nowy skrypt `SendStatePushNotification(newValue: number)`:

```
-- SendStatePushNotification(newValue: number)
GateHttp->StatePushNotification->SetQueryStringParams("val="..newValue)
GateHttp->StatePushNotification->SendRequest()
```

Aby poinformować klienta o nowym statusie należy wywołać skrypt podając jako parametr nową wartość cechy `Value`. Najlepiej zrobić to w zdarzeniu `OnValueChange` obiektu Relay. Ponieważ przypisaliśmy krok wcześniej wartość do zmiennej użytkownika `GateHttp->RelayValueOnGateHttp`, możemy jej użyć, aby uniknąć niepotrzebnego ponownego przesyłanie tej wartości. Zatem przypisanie wyglądać będzie następująco:

`GateHttp->SendStatePushNotification(GateHttp->RelayValueOnGateHttp)`

Należy pamiętać, że kopiowanie wartości do cechy `RelayValueOnGateHttp` musi nastąpić wcześniej.

Od teraz za każdym razem, gdy wartość cechy `Value` obiektu Relay się zmieni automatycznie zostanie wysłana notyfikacja z nową wartością.

Wybrana metoda przesyła nową wartość jako parametr URL, ale można oczywiście sformatować odpowiedź w dowolny sposób i przesłać w ciele wiadomości ustawiając wartość za pomocą metody `SetRequestBody(value)`.

> **Uwaga!** Należy pamiętać, że Gate Http otwiera nieograniczone możliwości kooperacji z systemem i można za jego pomocą wykonać dowolną operację, nawet szkodliwą. Dlatego ważne jest, aby konfiguracja Gate Http była starannie przemyślana i wykonana z najwyższą dbałością.

5. Przywracanie ustawień fabrycznych - *Hard Reset*

Uruchomienie funkcji *Hard Reset* na module GATE Http powoduje:

- Usunięcie zapisanej konfiguracji;
- Sformatowanie partycji pamięci flash;
- Usunięcie wszystkich utworzonych obiektów LUA;
- Utratę komunikacji pomiędzy OM / HM a modułem Gate.

W celu przywrócenia ustawień fabrycznych funkcją *Hard Reset* należy wykonać następujące czynności (zgodnie z podaną kolejnością):

- Odłączyć zasilanie od modułu Gate;
- Nacisnąć i przytrzymać przycisk *Reset* na module (przycisk znajduje się pod dolną zaślepką modułu);
- Podłączyć zasilanie do modułu Gate;
- Trzymać wciśnięty przycisk *Reset* przez co najmniej 10 sekund - podczas resetu dioda zielona będzie świecić światłem ciągłym. Prawidłowe wykonanie resetu zostanie potwierdzone 3-krotnym mrugnięciem diody zielonej.
- Po upływie 10 sekund zwolnić przycisk *Reset*
- Odczekać około 60 sekund aż do momentu, gdy na module diody - zielona oraz czerwona - będą mrugać naprzemiennie (tryb *Emergency*)

Po wykonaniu procedury na module zostanie wyczyszczona konfiguracja, natomiast sam moduł przestanie być widoczny (brak odpowiedzi na *Keep-Alive*) w projekcie z poziomu Object Managera. Aby ponownie przywrócić moduł, należy wykonać CLU Discovery a następnie wysłać konfigurację.

6. Parametry konfiguracyjne

> **Uwaga!** Opisana funkcjonalność oraz integracja jest dostępna dla **GRENTON GATE HTTP, DIN, Eth (INT-211-E-01)** posiadający **firmware 1.1.0-2034C lub wyższy!**

A. Obiekt GATE

CECHY

Nazwa	Opis
`Uptime`	Czas pracy urządzenia od ostatniego resetu (w sekundach)
`ClientReportInterval`	Okres raportowania o zmianach cech
`Date`	Aktualna data
`Time`	Aktualny czas (hh:mm:ss)
`LocalTime`	Aktualny lokalny znacznik czasu
`TimeZone`	Strefa czasowa
`UnixTime`	Aktualny uniksowy znacznik czasu
`FirmwareVersion`	Wersja oprogramowania Gate
`UseCloud`	Określa czy GATE łączy się do chmury
`CloudConnection`	Określa status połączenia GATE z chmurą
`NTPTimeout`	Czas oczekiwania na odpowiedź z serwera NTP
`UseNTP`	Określa czy GATE używa NTP
`PrimaryDNS`	Preferowany serwer DNS
`SecondaryDNS`	Alternatywny serwer DNS

METODY

Nazwa	Opis
`SetDateTime`	Ustawia datę i czas
`StartConsole`	Uruchamia konsolę Lua
`StartConsoleOnReboot`	Uruchamia konsolę Lua przy kolejnym uruchomieniu modułu
`SetClientReportInterval`	Ustawia okres raportowania o zmianach cech
`SetPrimaryDNS`	Ustawia cechę PrimaryDNS
`SetSecondaryDNS`	Ustawia cechę SecondaryDNS

ZDARZENIA

Nazwa	Opis
`OnInit`	Zdarzenie wywoływane jednorazowo w momencie inicjalizacji urządzenia

B. Obiekt HttpRequest

> **Uwaga!** Cechy opisane jako nieustawialne są cechami zawierającymi odpowiedzi. Wartości początkowe tych cech należy pozostawić niezmienione. Wszelkie operacje na tych zmiennych należy wykonywać na skryptach (oraz zmiennych lokalnych).

CECHY

Nazwa	Opis
`Host`	Adres hosta
`Path`	Ścieżka zapytania
`QueryStringParams`	Parametry zapytania. \z oznacza brak
`Method`	Typ metody wysyłanej w zapytaniu np. GET, POST
`Timeout`	Dopuszczalny czas odpowiedzi
`RequestType`	Typ zawartości wysyłanego zapytania. Definiuje parametr *content-type* w nagłówku zapytania. W zależności od wybranego typu zawartość cechy `RequestBody` jest odpowiednio serializowana: `0 - None` - niezdefiniowany. W nagłówku nie jest wysyłane content-type. Zawartość cechy `RequestBody` nie jest serializowana. `1 - Text` - *content-type: text/plain*. Zawartość cechy `RequestBody` nie jest serializowana. `2 - JSON` - *content-type: application/json*. Zawartość cechy `RequestBody` jest serializowana do formatu JSON. `3- XML` - *content-type: text/xml*. Zawartość cechy `RequestBody` jest serializowana do formatu XML. `4 - FormData` - *content-type: application/x-www-form-urlencoded*. Zawartość cechy `RequestBody` jest serializowana do tabeli. `5 - Other` - typ zawartości (*content-type*) jest inny niż wbudowany. Typ można zdefiniować umieszczając go w nagłówku (cecha `RequestHeaders`). Zawartość nie jest serializowana.
`ResponseType`	Typ oczekiwanej odpowiedzi. Definiuje parametr *Accept* w nagłówku zapytania. W zależności od wybranego typu zawartość otrzymanej odpowiedzi (cechy `ResponseBody`) jest odpowiednio parsowana do tabeli: `0 - None` - parametr *Accept* nie jest wysyłany w nagłówku wysyłanego zapytania. Odpowiedź (cecha `ResponseBody`) nie jest parsowana. `1 - Text` - *Accept: text/plain*. Odpowiedź (cecha `ResponseBody`) nie jest parsowana. `2 - JSON` - *Accept: application/json*. Odpowiedź (cecha `ResponseBody`) jest parsowana z JSON. `3 - XML` - *Accept: text/xml*. Odpowiedź (cecha `ResponseBody`) jest parsowana z XML. `4 - FormData` - *Accept: application/x-www-form-urlencoded*. Odpowiedź (cecha ResponseBody) jest parsowana. `5 - Other` - parametr *Accept* nagłówka jest inny niż wbudowany. Parametr można zdefiniować umieszczając go w nagłówku (cecha `RequestHeaders`).
`RequestHeaders`	Dodatkowe nagłówki zapytania HTTP. `\z` oznacza brak zawartości.
`RequestBody`	Zawartość wiadomości wysyłanej w zapytaniu. `\z` oznacza brak zawartości

Nazwa	Opis
`ResponseBody`	Zawartość wiadomości otrzymanej po wysłaniu zapytania. (cecha wykorzystywana do odczytu w skryptach - nieustawialna)
`StatusCode`	Status odpowiedzi HTTP

METODY

Nazwa	Opis
`SendRequest`	Wysyła zapytanie
`AbortRequest`	Przerywa obsługę zapytania
`Clear`	Usuwa treść zapytania
`SetHost`	Ustawia adres hosta
`SetPath`	Ustawia ścieżkę zapytania
`SetQueryStringParams`	Ustawia parametry zapytania
`SetMethod`	Ustawia metodę zapytania
`SetTimeout`	Ustawia dopuszczalny czas odpowiedzi
`SetRequestType`	Ustawia typ zawartości wysyłanego zapytania (content-type)
`SetResponseType`	Ustawia typ oczekiwanej odpowiedzi na zapytanie
`SetRequestHeaders`	Ustawia dodatkowe nagłówki HTTP w zapytaniu
`SetRequestBody`	Ustawia zawartość wiadomości w zapytaniu

ZDARZENIA

Nazwa	Opis
`OnRequestSent`	Zdarzenie wywoływane w momencie wysłania zapytania
`OnResponse`	Zdarzenie wywoływane w momencie otrzymania odpowiedzi

C. Obiekt HttpListener

> **Uwaga!** Cechy opisane jako nieustawialne są cechami zawierającymi odpowiedzi. Wartości początkowe tych cech należy pozostawić niezmienione. Wszelkie operacje na tych zmiennych należy wykonywać na skryptach (oraz zmiennych lokalnych)

CECHY

Nazwa	Opis
`Path`	Ścieżka zapytania
`Method`	Typ metody otrzymanej w zapytaniu np. `GET` , `POST`
`QueryStringParams`	Zwraca parametry zapytania HTTP (cecha wykorzystywana do odczytu w skryptach - nieustawialna)
`RequestType`	Typ otrzymanego zapytania. W zależności od wybranego typu, zawartość otrzymanego zapytania (cechy `RequestBody`) jest odpowiednio parsowana do tabeli: `0 - None` - Odpowiedź nie jest parsowana. `1 - Text` - Odpowiedź nie jest parsowana. `2 - JSON` - Odpowiedź jest parsowana z JSON. `3 - XML` - Odpowiedź jest parsowana z XML. `4 - FormData` - Odpowiedź jest parsowana. `5 - Other` - Odpowiedź nie jest parsowana. Cecha `RequestBody` zwraca treść zapytania HTTP (cecha wykorzystywana do odczytu w skryptach - nieustawialna).
`RequestBody`	Zwraca treść zapytania HTTP (cecha wykorzystywana do odczytu w skryptach - nieustawialna)
`ResponseType`	Typ zawartości wysłanej odpowiedzi na zapytanie. Definiuje parametr *content-type* w nagłówku odpowiedzi. W zależności od wybranego typu, zawartość cechy `ResponseBody` jest odpowiednio serializowana: `0 - None` - niezdefiniowany. W nagłówku nie jest wysyłane *content-type*. Zawartość nie jest serializowana. `1 - Text` - *content-type: text/plain*. Zawartość nie jest serializowana. `2 - JSON` - *content-type: application/json*. Zawartość `RequestBody` jest serializowana do formatu JSON. `3 - XML` - *content-type: text/xml*. Zawartość `RequestBody` jest serializowana do formatu XML. `4 - FormData` - *content-type: application/x-www-form-urlencoded*. Zawartość `RequestBody` jest serializowana. `5 - Other` - parametr *Accept* nagłówka jest inny niż wbudowany. Parametr można zdefiniować umieszczając go w nagłówku (cecha `RequestHeaders`).
`ResponseBody`	Zwraca treść odpowiedzi HTTP (cecha wykorzystywana do odczytu w skryptach).

Nazwa	Opis
`StatusCode`	Status wysyłanej odpowiedzi HTTP. Obsługiwane statusy: `200` - OK `201` - Utworzono `202` - Przyjęto `204` - Brak zawartości `205` - Przywróć zawartość `400` - Nieprawidłowe zapytanie `403` - Zabroniony `404` - Nie znaleziono `405` - Niedozwolona metoda `406` - Niedozwolone `408` - Koniec czasu oczekiwania na żądanie `409` - Konflikt `410` - Zniknął (usunięto)

METODY

Nazwa	Opis
`SendResponse`	Wysyła odpowiedź na zapytanie
`Clear`	Usuwa treść odpowiedzi
`SetPath`	Ustawia ścieżkę zapytania
`SetResponseType`	Ustawia typ oczekiwanej odpowiedzi na zapytanie
`SetResponseBody`	Ustawia treść odpowiedzi
`SetStatusCode`	Ustawia status odpowiedzi

ZDARZENIA

Nazwa	Opis
`OnRequest`	Zdarzenie wywoływane w momencie otrzymania zapytania

D. Obiekt Timer

CECHY

Nazwa	Opis
`Time`	Zliczany czas (w ms)
`Mode`	Tryb pracy timera: 0 – zliczenie w dół (countdown), 1 – cykliczny (interval)
`State`	Aktualny stan pracy timera: 0 – zatrzymany (stopped), 1 – liczy (counting)

METODY

Nazwa	Opis
`SetTime`	Ustawia czas timera (w ms)
`SetMode`	Ustawia tryb pracy: 0 – zliczenie w dół (countdown), 1 – cykliczny (interval)
`Start`	Uruchamia timer
`Stop`	Zatrzymuje timer

ZDARZENIA

Nazwa	Opis
`OnTimer`	Zdarzenie wywoływane przy zliczeniu timera
`OnStart`	Zdarzenie wywoływane przy uruchomieniu timera
`OnStop`	Zdarzenie wywoływane przy zatrzymaniu timera

XVI. Moduł DALI Controller

> **Uwaga!** Opisana funkcjonalność jest dostępna dla **GRENTON DALI Controller DIN, Eth (INT-202-D-01)** posiadający firmware **1.1.11 (build 2048)** lub wyższy.

> **Uwaga!** DALI Controller dostępny jest dla Object Managera w wersji **1.3.5 (build 204201)** i wyższej oraz dla CLU z firmware **5.06.04 (build 2050)** i wyższym.

1. Informacje ogólne

Moduł DALI Controller pełni rolę urządzenia nadrzędnego (master), zgodnie ze standardem DALI umożliwia obsługę 64 stateczników - Control Gears, podłączonych do magistrali DALI.

> **Uwaga!** Maksymalna ilość stateczników (obiektów DALI_GEAR) przypisanych do jednego CLU Z-Wave wynosi 128.

DALI Controller umożliwia sterowanie wszystkimi urządzeniami sterującymi światłem w zakresie zdefiniowanym przez normę PN-EN 62386-102, oraz rozszerzenie DT8.

Moduł pozwala na sterowanie pojedynczymi statecznikami, jak i również sterowanie za pomocą grup, każdy statecznik może zostać przypisany do 16 grup. Dzięki temu znacznie łatwiej jest uporządkować sterowanie oświetleniem oraz tworzyć zaawansowane scenariusze sterowania.

2. Konfiguracja modułu

> **Uwaga!** Przed rozpoczęciem jakiejkolwiek pracy z modułem DALI Controller wymagana jest aktualizacja bazy interfejsów!

Sygnalizacja diodami LED modułu

- Dioda niebieska sygnalizuje napięcie na magistrali DALI,
- Dioda zielona wskazuje obecny stan modułu:
  - Świeci ciągle – brak konfiguracji stateczników na module, należy przeprowadzić DALI Discovery,
  - Miga w interwale 200 ms – DALI Discovery, następuje wyszukiwanie stateczników podłączonych do magistrali DALI oraz nadawanie im adresów lokalnych,
  - Miga w interwale 1 s – konfiguracja stateczników znajduje się na module.

Dodanie modułu do projektu

Po wykonaniu procesu CLU Discovery w projekcie pojawiają się dwa obiekty:

- DALI_MASTER - główny obiekt służący do zarządzania konfiguracją modułu,
- AnalogIN - obiekt służący do monitorowania napięcia na magistrali systemowej.

A. Adresowanie stateczników

Konfigurację modułu należy rozpocząć od adresowania stateczników DALI podłączonych do magistrali. DALI Controller umożliwia zastosowanie dwóch typów adresowania: w pełni automatyczne lub ręczne.

Adresowanie automatyczne umożliwia adresowania całej instalacji jednym kliknięciem, za pomocą procesu DALI Discovery.

- W obiekcie DALI_MASTER po przejściu do zakładki `Sterowanie`, należy wywołać metodę `ResetGear (Broadcast)` a następnie metodę `DALI_Discovery`,
- Wywołanie metody inicjuje automatyczne adresowanie wszystkich stateczników na magistrali, które będą otrzymywały adresy lokalne z zakresu 0 do 63. Przydzielenie adresu zostanie potwierdzone przez zaświecenie danej oprawy na czas 300 ms. Należy mieć na uwadze, że po rozpoczęciu adresowania skasowane zostaną wszystkie dotychczasowe adresy. W trakcie DALI Discovery, adresy są przydzielane statecznikom w sposób losowy, zatem adresy nie będą stanowiły odzwierciedlenia w uporządkowanym montażu opraw,
- W czasie trwania DALI Discovery:
    - Zielona dioda na module DALI Controller miga w interwale 200 ms,
    - Cecha wbudowana `State` obiektu DALI_MASTER przyjmuje wartość 1.

Czas trwania DALI Discovery zależy od ilości stateczników (może trwać nawet do kilku minut dla maksymalnej ilość urządzeń).

> **Uwaga!** Podczas DALI Discovery nie należy wykonywać żadnych operacji na module DALI Controller!

Adresowanie ręczne umożliwia adresowanie pojedynczych stateczników za pomocą metody `SetLocalAddress`. Pomocne w przypadku, gdy po DALI Discovery statecznik nie zostanie odnaleziony, nastąpi zdublowanie adresu lub zależy nam na określonej kolejności adresów zgodnej z porządkiem montażu.

W obiekcie DALI_MASTER po przejściu do zakładki `Sterowanie`, wywołanie metody `SetLocalAddress` z parametrem `FindGear` ustawionym:

- `WithoutLocalAddress` - proces adresowania dla urządzenia nie posiadającego nadanego adresu,
    - `Address` - nowy niezajęty adres, który zostanie nadany urządzeniu,
- `WithLocalAddress` - proces adresowania dla urządzenia o podanym adresie,
    - `Address` - nowy niezajęty adres, który zostanie nadany urządzeniu,
- Przydzielenie adresu w obu przypadkach zostanie potwierdzone przez zaświecenie danej oprawy na czas 300 ms,
- W czasie trwania `SetLocalAddress`:
    - Zielona dioda na module DALI Controller miga w interwale 200 ms,
    - Cecha wbudowana `State` obiektu DALI_MASTER przyjmuje wartość 1.

> **Uwaga!** Podczas działania metody `SetLocalAddress` nie należy wykonywać żadnych operacji na module DALI Controller!

Po zakończonym procesie DALI Discovery

- Zielona dioda na module DALI Controller miga w interwale 1 s (odnalezione stateczniki) lub świeci ciągle (brak odnalezionych stateczników),
- Cecha wbudowana `State` obiektu DALI_MASTER przyjmuje wartość:

    - 3 - w przypadku, gdy urządzenia DALI zostaną odnalezione,
    - 0 - w przypadku braku znalezionych urządzeń,
- Cecha wbudowana `NumberOfGear` obiektu DALI_MASTER zwraca ilość poprawnie odnalezionych i zaadresowanych urządzeń,
- Generowane jest zdarzenie `OnDALI_DiscoveryCompleated`.

Operacje możliwe na urządzeniach po zakończonym DALI Discovery

Za pomocą metod obiektu DALI_MASTER możemy:

- Zweryfikować urządzenie zgłaszające się pod danym adresem - metoda `Identify`,
- Zrestartować urządzenie pod danym adresem - metoda `ResetGear`,
- Ustawić wartość oprawy dla urządzenia pod danym adresem - metoda `SetDAPCValue`.

B. Dodanie stateczników do projektu

Po zakończenia procesu adresowania stateczników metodami `DALI_Discovery` oraz `SetLocalAddress` należy przeprowadzić CLU Discovery:

- Do projektu dodawane są nowe obiekty GEAR reprezentujące każde urządzenie (adres) DALI poprawnie odnalezione i dodane podczas procesu adresowania,
- Cecha wbudowana `GearAddresses` obiektu DALI_MASTER zwraca numery adresów w zakresie 0 - 63, zajętych przez urządzenia DALI
- Obiekty GEAR występują w wersji DALI_GEAR i DALI_GEAR_DT8 - Device Type 8:
    - DALI_GEAR - wszystkie stateczniki z podstawowymi metodami sterowania,
    - DALI_GEAR_DT8 - stateczniki umożliwiające sterowanie kolorami (tryb sterowania RGBWA) lub temperaturą koloru (tryb sterowania Tc).

> **Uwaga!** Dla prawidłowego działania konfiguracji i obiektów GEAR, CLU Discovery należy przeprowadzić po każdej zmianie w adresacji stateczników!

C. Sterowanie statecznikami

Sterowanie pojedynczym statecznikiem odbywa się z poziomu danego obiektu DALI_GEAR / DALI_GEAR_DT8 dostępnymi metodami lub za pomocą metod obiektu DALI_MASTER (dokładne funkcjonalności znajdują się w opisie poszczególnych obiektów).

Sterowanie grupami stateczników odbywa się z poziomu obiektu DALI_MASTER metodami `SetGroupDAPCValue`, `GroupSwitchOn`, `GroupSwitchOff`. Aby możliwe było sterowanie daną grupą urządzeń, należy:

- Dla pożądanych obiektów GEAR ustawić wartość cechy wbudowanej `Group`. Każdy obiekt można przypisać do 16 grup z zakresu 1 - 16, kolejne grupy podawane po przecinku,
- Po przypisaniu obiektów do grup, należy wysłać konfigurację na CLUZ,
- Po wysłaniu konfiguracji następuje nadawanie grup przez moduł DALI Controller. Cecha wbudowana `State` obiektu DALI_MASTER przyjmuje wartość 4. Czas trwania procesu jest zależny od ilości urządzeń dla których została zmieniona wartość cechy `Group`, może trwać do 60 sekund,
- Po poprawnym nadaniu grup, cecha wbudowana `State` obiektu DALI_MASTER przyjmuje wartość 3.

> **Uwaga!** Podczas nadawania grup (po restarcie CLUZ / wysłaniu konfiguracji) nie ma możliwości sterowania obiektami!

D. RampTime

DALI Controller realizuje obsługę płynnej zmiany wartości `DAPCValue` za pomocą parametru `RampTime`, w sposób logarytmiczny:

RampTime	Minimalny czas ustawiania [s]	Nominalny czas ustawiania [s]	Maksymalny czas ustawiania [s]
1	0,6	0,7	0,8
2	0,9	1,0	1,1
3	1,3	1,4	1,6
4	1,8	2,0	2,2
5	2,5	2,8	3,1
6	3,6	4,0	4,4
7	5,1	5,7	6,2
8	7,2	8,0	8,8
9	10,2	11,3	12,4
10	14,4	16,0	17,6
11	20,4	22,6	24,9
12	28,8	32,0	35,2
13	40,7	45,3	49,8
14	57,6	64,0	70,4
15	81,5	90,5	99,6

3. Obiekty

A. DALI_MASTER

CECHY

Nazwa	Opis
`State`	`0` - brak konfiguracji stateczników `1` - DALI Discovery `3` - konfiguracja stateczników znajduje się na urządzeniu `4` - zapisywanie informacji o grupach
`NumberOfGear`	Liczba stateczników w konfiguracji urządzenia
`GearAddresses`	Adresy stateczników nadane podczas DALI Discovery. Wartość cechy odświeżana jest po restarcie systemu

METODY

Nazwa	Opis
`Identify`	Włącza oprawę na 2 sekundy
`ResetGear`	Resetuje statecznik
`SetLocalAddress`	Ustawia adres lokalny statecznika
`DALI_Discovery`	Wyszukiwanie stateczników podłączonych do magistrali DALI oraz nadawanie im adresów lokalnych. W momencie nadania adresu, dany statecznik zostaje załączony na 300 ms. Podczas `DALI_Discovery` nie należy wykonywać operacji na urządzeniu!
`SetDAPCValue`	Ustawia wartość z jaką świeci oprawa. Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`SetGroupDAPCValue`	Ustawia wartość z jaką świeci oprawa dla podanej grupy. Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`GroupSwitchOn`	Włącza oprawy dla podanej grupy. Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`GroupSwitchOff`	Wyłącza oprawy dla podanej grupy. Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]

ZDARZENIA

Nazwa	Opis
`OnDALI_DiscoveryCompleated`	Zdarzenie wywoływane po zakończeniu wyszukiwania stateczników i nadawaniu im adresów lokalnych

B. DALI_GEAR

CECHY

Nazwa	Opis
`Address`	Adres statecznika
`Group`	Numery grup do których należy statecznik, kolejne grupy z zakresu 1-16 podawane po przecinku. 0 - brak przynależności do grupy
`DAPCValue`	Wartość z jaką świeci oprawa

METODY

Nazwa	Opis
`Identify`	Włącza oprawę na 2 sekundy
`SetDAPCValue`	Ustawia wartość z jaką świeci oprawa. Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`Switch`	Zmienia stan oprawy na przeciwny (0 / 254). Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`SwitchOn`	Włącza oprawę. Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`SwitchOff`	Wyłącza oprawę. Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`Hold`	Realizuje funkcję rozjaśniania / ściemniania oprawy
`HoldUp`	Realizuje funkcję rozjaśniania oprawy
`HoldDown`	Realizuje funkcję ściemniania oprawy

ZDARZENIA:

Nazwa	Opis
`OnDAPCValueChange`	Zdarzenie wywoływane przy zmianie wartości `DAPCValue`
`OnSwitchOn`	Zdarzenie wywoływane przy zmianie wartości `DAPCValue` z 0 na wartość większą
`OnSwitchOff`	Zdarzenie wywoływane przy zmianie wartości `DAPCValue` na 0

C. DALI_GEAR_DT8

CECHY

Nazwa	Opis
`Address`	Adres statecznika
`Group`	Numery grup do których należy statecznik, kolejne grupy z zakresu 1-16 podawane po przecinku. 0 - brak przynależności do grupy
`DAPCValue`	Wartość z jaką świeci oprawa
`HSVValue`	Wartość jasności według modelu HSV (zakres 0.00-1.00). Cecha nie pobiera rzeczywistego stanu jasności oprawy! Ustawiana na podstawie wywołanej metody `SetHSVValue`.
`HSVSaturation`	Wartość nasycenia barwy według modelu HSV (0.00-1.00). Cecha nie pobiera rzeczywistego stanu nasycenia barwy oprawy! Ustawiana na podstawie wywołanej metody `SetHSVSaturation`.
`HSVHue`	Wartość koloru barwy według modelu HSV (0-360). Cecha nie pobiera rzeczywistego stanu koloru barwy oprawy! Ustawiana na podstawie wywołanej metody `SetHSVHue`.

METODY

Nazwa	Opis
`Identify`	Włącza oprawę na 2 sekundy
`SetDAPCValue`	Ustawia wartość z jaką świeci oprawa. Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`Switch`	Zmienia stan oprawy na przeciwny (0 / 254). Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`SwitchOn`	Włącza oprawę. Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`SwitchOff`	Wyłącza oprawę. Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`Hold`	Realizuje funkcję rozjaśniania / ściemniania oprawy
`HoldUp`	Realizuje funkcję rozjaśniania oprawy
`HoldDown`	Realizuje funkcję ściemniania oprawy
`SetHSVValue`	Ustawia wartość jasności według modelu HSV (zakres 0.00-1.00). Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`SetHSVSaturation`	Ustawia wartość nasycenia według modelu HSV (zakres 0.00-1.00). Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`SetHSVHue`	Ustawia wartość barwy według modelu HSV (zakres 0-360). Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`SetRGBValue`	Ustawia wartość kanałów R (Red), G (Green), B (Blue). Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`SetWAFValue`	Ustawia wartość kanału W (White), oraz parametrów A (Amber) i F (Freecolour). Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]
`SetColourTemperature`	Ustawia wartość temperatury koloru, gdzie 0 - fizyczne minimum, 100 - fizyczne maksimum. Parametr `RampTime` ustawiany w skali logarytmicznej 0.8 - 90 [s]

ZDARZENIA:

Nazwa	Opis
`OnDAPCValueChange`	Zdarzenie wywoływane przy zmianie wartości `DAPCValue`
`OnSwitchOn`	Zdarzenie wywoływane przy zmianie wartości `DAPCValue` z 0 na wartość większą
`OnSwitchOff`	Zdarzenie wywoływane przy zmianie wartości `DAPCValue` na 0

D. AnalogIN

CECHY

Nazwa	Opis
`Value`	Wartość napięcia zasilania
`Value%`	Wartość napięcia zasilania jako procent wartości maksymalnej `MaxValue`
`Sensitivity`	Czułość - minimalna zmiana wartości napięcia zasilania, która wywołuje zdarzenia `OnValueChange`, `OnValueLower` lub `OnValueRise`
`MinValue`	Wartość minimalna napięcia zasilania po przekroczeniu której generowane jest zdarzenie `OnOutOfRange`
`MaxValue`	Wartość maksymalna napięcia zasilania po przekroczeniu której generowane jest zdarzenie `OnOutOfRange`

METODY

Nazwa	Opis
`SetSensitivity`	Ustawia czułość pomiaru napięcia zasilania
`SetMinValue`	Ustawia wartość `MinValue`
`SetMaxValue`	Ustawia wartość `MaxValue`

ZDARZENIA:

Nazwa	Opis
`OnValueChange`	Zdarzenie wywoływane przy zmianie wartości napięcia zasilania
`OnValueLower`	Zdarzenie wywoływane przy zmianie wartości napięcia zasilania na niższą (zbocze opadające)
`OnValueRise`	Zdarzenie wywoływane przy zmianie wartości napięcia zasilania na wyższą (zbocze narastające)
`OnOutOfRange`	Zdarzenie wywoływane gdy wartość napięcia zasilania znajduje się poza wyznaczonym zakresem (`MinValue`; `MaxValue`)
`OnInRange`	Zdarzenie wywoływane gdy wartość napięcia zasilania powróci do wyznaczonego zakresu (`MinValue`; `MaxValue`)

XVII. Moduły Z-Wave

Niniejszy rozdział przedstawia opis zakresu wsparcia modułów Z-Wave innych producentów, które są dostępne w systemie Grenton.

> **Uwaga!** Pełna lista urządzeń jest dostępna na stronie https://support.grenton.pl/pl/support/solutions w artykule '*Jakie bezprzewodowe moduły Z-Wave są obsługiwane?*'

1. Fibaro UBS

Wersja modułu: FGBS-001 v2.1.

1.1. Informacje ogólne

Moduł Z-Wave Fibaro UBS posiada dwa wejścia bezpotencjałowe. Umożliwia odczyt wartości nawet z czterech czujników 1-Wire. Ponadto daje możliwość zmiany parametrów konfiguracyjnych (interfejs konfiguracyjny Fibaro).

> **Uwaga!** Dodawanie/usuwanie odbywa się przez trzykrotne kliknięcie przycisku w module podczas inclusion/exclusion.

1.2. Obiekty

A. ZWAVE_DIN

Wejścia bezpotencjałowe.

CECHY

Nazwa	Opis
`Value`	Zwraca stan wejścia
`HoldDelay`	Czas, po którym wciśnięcie i przytrzymanie przycisku wyzwoli zdarzenie `OnHold`
`HoldInterval`	Odstęp cykliczny (w ms), po jakim podczas trzymania przycisku wyzwalane są kolejne zdarzenia `OnHold`

METODY

Nazwa	Opis
`SetHoldDelay`	Ustawia wartość `HoldDelay`
`SetHoldInterval`	Ustawia wartość `HoldInterval`

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w przypadku zmiany stanu na przeciwny
`OnSwitchOn`	Zdarzenie wywoływane w momencie ustawienia stanu wysokiego na wejściu
`OnSwitchOff`	Zdarzenie wywoływane w momencie ustawienia stanu niskiego na wejściu
`OnShortPress`	Zdarzenie wywoływane po naciśnięciu przycisku na okres 500-2000ms
`OnLongPress`	Zdarzenie wywoływane po naciśnięciu przycisku na okres 2000-5000ms
`OnHold`	Zdarzenie wywoływane, gdy wejście jest w stanie wysokim, pierwszy raz po upłynięciu czasu `HoldDelay`, a następnie cyklicznie co wartość `HoldInterval`
`OnClick`	Zdarzenie wywoływane po naciśnięciu przycisku na czas krótszy niż 500ms

B. ZWAVE_1W_SENSOR

Obiekt odpowiada za czujnik 1-Wire. Dla każdego czujnika tworzony jest osobny obiekt. Do modułu UBS Fibaro można podłączyć maksymalnie 4 czujniki 1-Wire (DS18B20).

Obiekty ZWAVE_1W_SENSOR są zawsze dodawane wraz z dodaniem modułu Fibaro UBS do CLU/projektu w OM, niezależnie od ilości podłączonych czujników. O tym, czy dany czujnik jest podłączony, informuje cecha Discovered - zwracająca informację, czy podczas Discovery pod dany obiekt zgłosił się czujnik 1-Wire i jest podłączony do modułu UBS.

Przy podłączeniu lub odłączeniu czujników 1-Wire, należy usunąć a następnie ponownie dodać moduł UBS do modułu CLU Z-Wave. Moduł Fibaro UBS zgłosi się z nowym numerem seryjnym - istnieje możliwość przepisania konfiguracji obiektów (automatyczna lub ręczna). Po ponownym dodaniu czujników może nastąpić ponowne przeindeksowanie kolejności czujników do obiektów ZW_1W_SENSOR.

Moduł Fibaro UBS dla czujnika 1-Wire nie zwraca informacji, czy podczas działania systemu czujnik został rozłączony - przechowywana jest ostatnia pobrana wartość, dlatego niezalecane jest wykorzystywanie tych czujników jako źródło regulacji temperatury.

W momencie zwarcia na magistrali 1-Wire, wszystkie czujniki podłączone do modułu Fibaro UBS (dostępne/widoczne w OM) zwracają wartość 0.00 - dlatego przy dłuższym (nieplanowanym) wystąpieniu tej wartości należy sprawdzić poprawność podłączenia magistrali 1-Wire.

CECHY

Nazwa	Opis
`Value`	Wartość wejścia
`MinValue`	Minimalna wartość wejścia
`MaxValue`	Maksymalna wartość wejścia
`Discovered`	Informacja zwracana podczas CLU Discovery o podłączeniu czujnika do modułu

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane przy zmianie wartości wyjścia
`OnRise`	Zdarzenie wywoływane po przekroczeniu górnego progu histerezy (zbocze rosnące)
`OnLower`	Zdarzenie wywoływane po przekroczeniu dolnego progu histerezy (zbocze opadające)
`OnOutOfRange`	Zdarzenie wywoływane, gdy wartość na wyjściu znajduje się poza wyznaczonym zakresem (`MinValue` : `MaxValue`)
`OnInRange`	Zdarzenie wywoływane przy powrocie wartości do przedziału wewnątrz wartości progowych (`MinValue` : `MaxValue`)

C. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów oraz komunikacji z modułem w sieci Z-Wave. Umożliwia ustawienie zaawansowanych parametrów konfiguracyjnych danego modułu (określone indywidualnie w instrukcji).

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` – komunikacja z modułem nie jest zablokowana, `1` – zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie – jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)
`Register`	Numer rejestru (parametru) konfiguracyjnego, który ostatnio został odczytany/ustawiony za pomocą dostępnych metod
`Value`	Wartość rejestru (parametru) konfiguracyjnego

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** *Cecha `RemoveBan` nie jest jednoznaczne z ponowną poprawną komunikacją z modułem - umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji
`Set`	Ustawia wartość danego rejestru (parametru) konfiguracyjnego: `1 - Register` (numer rejestru lub parametru), `2 - Value` (wartość rejestru bądź parametru), `3 - Size` (rozmiar wysyłanej wartości rejestru lub parametru - w bajtach)
`Get`	Pobiera wartość danego rejestru (parametru) konfiguracyjnego
`SetDefault`	Ustawia wartość domyślną dla danego rejestru (parametru) konfiguracyjnego

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

2. NEO Coolcam Motion Sensor (PIR)

Wersja modułu: NAS-PD01ZE HW: 66 FW: 3.80

2.1. Informacje ogólne

Moduł Z-Wave Neo Coolcam Motion Sensor umożliwia odczyt: stanu czujnika ruchu (PIR), poziomu oświetlenia oraz poziomu baterii. Ponadto daje możliwość ustawienia/odczytu czasu wybudzenia modułu.

> **Uwaga!** Dodawanie/usuwanie odbywa się przez trzykrotne kliknięcie przycisku w module Neo podczas inclusion/exclusion. Poprawnie przeprowadzony proces zostanie potwierdzony pięciokrotnym mrugnięciem diody.

2.2. Obiekty

A. BINARY_SENSOR

Obiekt umożliwia odczyt stanu czujnika ruchu.

CECHY

Nazwa	Opis
`Value`	Zwraca stan wejścia: **0** - brak naruszenia, **1** - naruszenie

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w przypadku zmiany stanu na przeciwny
`OnSwitchOn`	Zdarzenie wywoływane w momencie ustawienia stanu wysokiego na wejściu
`OnSwitchOff`	Zdarzenie wywoływane w momencie ustawienia stanu niskiego na wejściu

B. ANALOG_SENSOR

Obiekt umożliwia odczyt natężenia oświetlenia mierzonego w luksach.

CECHY

Nazwa	Opis
`Value`	Aktualna wartość sensora
`MinValue`	Wartość, poniżej której generowane jest zdarzenie `OnOutOfRange`
`MaxValue`	Wartość, powyżej której generowane jest zdarzenie `OnOutOfRange`

METODY

Nazwa	Opis
`SetMinValue`	Ustawia dolną wartość progową zdarzenia `OnOutOfRange`
`SetMaxValue`	Ustawia górną wartość progową zdarzenia `OnOutOfRange`

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane przy zmianie wartości sensora
`OnValueRaise`	Zdarzenie wywoływane przy zmianie wartości sensora na wyższą niż poprzednia
`OnValueDrop`	Zdarzenie wywoływane przy zmianie wartości sensora na niższą niż poprzednia
`OnOutOfRange`	Zdarzenie wywoływane przy przekroczeniu jednej z wartości progowych `MinValue` / `MaxValue`
`OnInRange`	Zdarzenie wywoływane przy powrocie wartości do przedziału wewnątrz wartości progowych (`MinValue` : `MaxValue`)

C. ZWAVE_BATTERY

Obiekt umożliwia odczyt stanu baterii. Odczyt odbywa się cyklicznie, co ustawiony czas, dla cechy `Interval` obiektu ZWAVE_WAKEUP (domyślnie 3600s).

CECHY

Nazwa	Opis
`BatteryLevel`	Poziom baterii modułu Z-Wave (w procentach)
`WarningLevel`	Poziom baterii, poniżej którego generowane są zdarzenia ostrzegawcze

METODY

Nazwa	Opis
`SetWarningLevel`	Ustawia poziom ostrzegawczy baterii modułu Z-Wave

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane przy zmianie wartości poziomu baterii
`OnLowBattery`	Zdarzenie wywoływane po wykryciu spadku poziomu baterii poniżej poziomu ostrzegawczego
`OnBatteryGood`	Zdarzenie wywoływane po wykryciu powrotu poziomu baterii do wartości powyżej poziomu ostrzegawczego

D. ZWAVE_WAKEUP

Obiekt umożliwia ustawienie oraz odczyt czasu wybudzenia bateryjnego modułu Z-Wave. Domyślna wartość ustawiana przez CLU wynosi 3600s (60 minut). Minimalna wartość to 300s (5 minut); maksymalna 16777200s (około 194 dni). Możliwe jest ustawienie wartości w kroku 60s (360s, 420s, 480s itd.)

CECHY

Nazwa	Opis
`Interval`	Czas samoczynnego wybudzania modułu Z-Wave z trybu uśpienia (w sekundach)
`LastWakeUp`	Czas ostatniego wybudzenia modułu Z-Wave z trybu uśpienia

METODY

Nazwa	Opis
`SetInterval`	Ustawia czas samoczynnego wybudzenia modułu Z-Wave z trybu uśpienia

ZDARZENIA

Nazwa	Opis
`OnWakeUp`	Zdarzenie wywoływane po wykryciu wybudzenia modułu Z-Wave z trybu uśpienia

E. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów oraz komunikacji z modułem w sieci Z-Wave. Umożliwia ustawienie zaawansowanych parametrów konfiguracyjnych danego modułu (określone indywidualnie w instrukcji).

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` - komunikacja z modułem nie jest zablokowana, `1` - zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie - jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)
`Register`	Numer rejestru (parametru) konfiguracyjnego, który ostatnio został odczytany/ustawiony za pomocą dostępnych metod
`Value`	Wartość rejestru (parametru) konfiguracyjnego **Uwaga!** *Parametr 2, 3, 5 oraz 8 odnosi się do asocjacji modułów, która nie jest wspierana przez system Grenton!* **Uwaga!** *Parametr 3 - zmiana wartości parametru nie powoduje wysyłania jej podczas wykrycia ruchu!* **Uwaga!** *Parametr 4 - poprawne ustawianie wartości parametru, jednakże sam moduł nie przestawia trybu pracy*! **Uwaga!** *Parametr 7 oraz 9 - poprawne ustawianie wartości parametru, jednakże ustawiona wartość nie została przetestowana ze względu na uszkodzony czujnik!* **Uwaga!** *Parametr 1, 6 - brak zauważalnych zmian w pracy modułu po zmianie wartości!* **Uwaga!** *Parametr 9 - mniejszy zakres ustawianych wartości (do 100 luxów)!* **Uwaga!** *W dokumentacji brak informacji o rejestrze numer 11 (Motion Event Report One Time Enable)!*

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` *nie jest jednoznaczne z ponowną poprawną komunikacją z modułem - umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji
`Set`	Ustawia wartość danego rejestru (parametru) konfiguracyjnego: `Register` (numer rejestru lub parametru), `Value` (wartość rejestru bądź parametru), `Size` (rozmiar wysyłanej wartości rejestru lub parametru - w bajtach) **Uwaga!** *Wywołanie metody* `Set` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!*
`Get`	Pobiera wartość danego rejestru (parametru) konfiguracyjnego **Uwaga!** *Wywołanie metody* `Get` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!*
`SetDefault`	Ustawia wartość domyślną dla danego rejestru (parametru) konfiguracyjnego **Uwaga**! *Wywołanie metody* `SetDefault` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!*

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

3. NEO Coolcam Door / Window Sensor

Wersja modułu: NAS-DS01Z

3.1. Informacje ogólne

Moduł Z-Wave Neo Coolcam Door/Window Sensor umożliwia odczyt: stanu kontaktronu (NC) oraz poziomu baterii. Ponadto daje możliwość ustawienia/odczytu czasu wybudzenia modułu.

> **Uwaga!** Dodawanie/usuwanie odbywa się przez trzykrotne kliknięcie przycisku w module Neo podczas inclusion/exclusion. Poprawnie przeprowadzony proces zostanie potwierdzony pięciokrotnym mrugnięciem diody.

3.2. Obiekty

A. BINARY_SENSOR

Obiekt umożliwia odczyt stanu otwarcia/zamknięcia kontaktronu.

CECHY

Nazwa	Opis
`Value`	Zwraca stan wejścia: **0** - zamknięcie, **1** - otwarcie

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w przypadku zmiany stanu na przeciwny
`OnSwitchOn`	Zdarzenie wywoływane w momencie ustawienia stanu wysokiego na wejściu
`OnSwitchOff`	Zdarzenie wywoływane w momencie ustawienia stanu niskiego na wejściu

B. ZWAVE_BATTERY

Obiekt umożliwia odczyt stanu baterii. Odczyt odbywa się cyklicznie, co ustawiony czas, dla cechy `Interval` obiektu `ZWAVE_WAKEUP`.

CECHY

Nazwa	Opis
`BatteryLevel`	Poziom baterii modułu Z-Wave (w procentach)
`WarningLevel`	Poziom baterii, poniżej którego generowane są zdarzenia ostrzegawcze

METODY

Nazwa	Opis
`SetWarningLevel`	Ustawia poziom ostrzegawczy baterii modułu Z-Wave

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane przy zmianie wartości poziomu baterii
`OnLowBattery`	Zdarzenie wywoływane po wykryciu spadku poziomu baterii poniżej poziomu ostrzegawczego
`OnBatteryGood`	Zdarzenie wywoływane po wykryciu powrotu poziomu baterii do wartości powyżej poziomu ostrzegawczego

C. ZWAVE_WAKEUP

Obiekt umożliwia ustawienie oraz odczyt czasu wybudzenia bateryjnego modułu Z-Wave. Domyślna wartość ustawiana przez CLU wynosi 3600s (60 minut). Minimalna wartość to 300s (5 minut); maksymalna 16777200s (około 194 dni). Możliwe jest ustawienie wartości w kroku 60s (360s, 420s, 480s itd.)

CECHY

Nazwa	Opis
`Interval`	Okres samoczynnego wybudzania modułu Z-Wave z trybu uśpienia (w sekundach)
`LastWakeUp`	Czas ostatniego wybudzenia modułu Z-Wave z trybu uśpienia

METODY

Nazwa	Opis
`SetInterval`	Ustawia okres samoczynnego wybudzenia modułu Z-Wave z trybu uśpienia

ZDARZENIA

Nazwa	Opis
`OnWakeUp`	Zdarzenie wywoływane po wykryciu wybudzenia modułu Z-Wave z trybu uśpienia

D. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów oraz komunikacji z modułem w sieci Z-Wave. Umożliwia ustawienie zaawansowanych parametrów konfiguracyjnych danego modułu (określone indywidualnie w instrukcji).

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` – komunikacja z modułem nie jest zablokowana, `1` – zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie – jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)
`Register`	Numer rejestru (parametru) konfiguracyjnego, który ostatnio został odczytany/ustawiony za pomocą dostępnych metod
`Value`	Wartość rejestru (parametru) konfiguracyjnego **Uwaga!** *Parametr 1 i 2 odnosi się do asocjacji modułów, która nie jest wspierana przez system Grenton!*

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` *nie jest jednoznaczne z ponowną poprawną komunikacją z modułem - umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji
`Set`	Ustawia wartość danego rejestru (parametru) konfiguracyjnego: `Register` (numer rejestru lub parametru), `Value` (wartość rejestru bądź parametru), `Size` (rozmiar wysyłanej wartości rejestru lub parametru - w bajtach) **Uwaga!** *Wywołanie metody* `Set` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!*
`Get`	Pobiera wartość danego rejestru (parametru) konfiguracyjnego **Uwaga!** *Wywołanie metody* `Get` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!***
`SetDefault`	Ustawia wartość domyślną dla danego rejestru (parametru) konfiguracyjnego **Uwaga**! *Wywołanie metody* `SetDefault` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!***

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

4. INFIBITY Motion Sensor (PIR) [NEO Coolcam]

Wersja modułu: NAS-PD01ZE HW: 66 FW: 3.80

4.1. Informacje ogólne

Moduł Z-Wave Infibity Motion Sensor umożliwia odczyt: stanu czujnika ruchu (PIR), poziomu oświetlenia, temperatury oraz poziomu baterii. Ponadto daje możliwość ustawienia/odczytu czasu wybudzenia modułu.

> **Uwaga!** Dodawanie/usuwanie odbywa się przez trzykrotne kliknięcie przycisku w module Infibity podczas inclusion/exclusion. Poprawnie przeprowadzony proces zostanie potwierdzony pięciokrotnym mrugnięciem diody.

4.2. Obiekty

A. BINARY_SENSOR

Obiekt umożliwia odczyt stanu czujnika ruchu.

CECHY

Nazwa	Opis
`Value`	Zwraca stan wejścia: **0** - brak naruszenia, **1** - naruszenie

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w przypadku zmiany stanu na przeciwny
`OnSwitchOn`	Zdarzenie wywoływane w momencie ustawienia stanu wysokiego na wejściu
`OnSwitchOff`	Zdarzenie wywoływane w momencie ustawienia stanu niskiego na wejściu

B. ANALOG_SENSOR

Obiekt umożliwia odczyt natężenia oświetlenia mierzonego w luxach (ANALOG_SENSOR1) oraz temperatury (ANALOG_SENSOR2).

CECHY

Nazwa	Opis
`Value`	Aktualna wartość sensora
`MinValue`	Wartość, poniżej której generowane jest zdarzenie `OnOutOfRange`
`MaxValue`	Wartość, powyżej której generowane jest zdarzenie `OnOutOfRange`

METODY

Nazwa	Opis
`SetMinValue`	Ustawia dolną wartość progową zdarzenia `OnOutOfRange`
`SetMaxValue`	Ustawia górną wartość progową zdarzenia `OnOutOfRange`

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane przy zmianie wartości sensora
`OnValueRaise`	Zdarzenie wywoływane przy zmianie wartości sensora na wyższą niż poprzednia
`OnValueDrop`	Zdarzenie wywoływane przy zmianie wartości sensora na niższą niż poprzednia
`OnOutOfRange`	Zdarzenie wywoływane przy przekroczeniu jednej z wartości progowych `MinValue` / `MaxValue`
`OnInRange`	Zdarzenie wywoływane przy powrocie wartości do przedziału wewnątrz wartości progowych (`MinValue` : `MaxValue`)

C. ZWAVE_BATTERY

Obiekt umożliwia odczyt stanu baterii. Odczyt odbywa się cyklicznie, co ustawiony czas, dla cechy `Interval` obiektu **`ZWAVE_WAKEUP`** (domyślnie 3600s).

CECHY

Nazwa	Opis
`BatteryLevel`	Poziom baterii modułu Z-Wave (w procentach)
`WarningLevel`	Poziom baterii, poniżej którego generowane są zdarzenia ostrzegawcze

METODY

Nazwa	Opis
`SetWarningLevel`	Ustawia poziom ostrzegawczy baterii modułu Z-Wave

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane przy zmianie wartości poziomu baterii
`OnLowBattery`	Zdarzenie wywoływane po wykryciu spadku poziomu baterii poniżej poziomu ostrzegawczego
`OnBatteryGood`	Zdarzenie wywoływane po wykryciu powrotu poziomu baterii do wartości powyżej poziomu ostrzegawczego

D. ZWAVE_WAKEUP

Obiekt umożliwia ustawienie oraz odczyt czasu wybudzenia bateryjnego modułu Z-Wave. Domyślna wartość ustawiana przez CLU wynosi 3600s (60 minut). Minimalna wartość to 300s (5 minut); maksymalna 16777200s (około 194 dni). Możliwe jest ustawienie wartości w kroku 60s (360s, 420s, 480s itd.)

CECHY

Nazwa	Opis
`Interval`	Okres samoczynnego wybudzania modułu Z-Wave z trybu uśpienia (w sekundach)
`LastWakeUp`	Czas ostatniego wybudzenia modułu Z-Wave z trybu uśpienia

METODY

Nazwa	Opis
`SetInterval`	Ustawia okres samoczynnego wybudzenia modułu Z-Wave z trybu uśpienia

ZDARZENIA

Nazwa	Opis
`OnWakeUp`	Zdarzenie wywoływane po wykryciu wybudzenia modułu Z-Wave z trybu uśpienia

E. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów oraz komunikacji z modułem w sieci Z-Wave. Umożliwia ustawienie zaawansowanych parametrów konfiguracyjnych danego modułu (określone indywidualnie w instrukcji).

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` – komunikacja z modułem nie jest zablokowana, `1` – zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie – jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)
`Register`	Numer rejestru (parametru) konfiguracyjnego, który ostatnio został odczytany/ustawiony za pomocą dostępnych metod
`Value`	Wartość rejestru (parametru) konfiguracyjnego **Uwaga!** *Parametr 2, 3, 5 oraz 8 odnosi się do asocjacji modułów, która nie jest wspierana przez system Grenton!* **Uwaga!** *Parametr 1, 6 oraz 7 – brak zauważalnych zmian w pracy modułu po zmianie wartości!* **Uwaga!** Parametr 9 – mniejszy zakres ustawianych wartości (do 100 luksów)!

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` *nie jest jednoznaczne z ponowną poprawną komunikacją z modułem - umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji
`Set`	Ustawia wartość danego rejestru (parametru) konfiguracyjnego: `Register` (numer rejestru lub parametru), `Value` (wartość rejestru bądź parametru), `Size` (rozmiar wysyłanej wartości rejestru lub parametru - w bajtach) **Uwaga!** *Wywołanie metody* `Set` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!*
`Get`	Pobiera wartość danego rejestru (parametru) konfiguracyjnego **Uwaga!** *Wywołanie metody* `Get` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!*
`SetDefault`	Ustawia wartość domyślną dla danego rejestru (parametru) konfiguracyjnego **Uwaga!** *Wywołanie metody* `SetDefault` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!*

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

5. INFIBITY Door/Window Sensor [NEO Coolcam]

Wersja modułu: NAS-DS01Z HW: 65 FW: 3.61

5.1. Informacje ogólne

Moduł Z-Wave Infibity Door/Window Sensor umożliwia odczyt: stanu kontaktronu (NC) oraz poziomu baterii. Ponadto daje możliwość ustawienia/odczytu czasu wybudzenia modułu.

> **Uwaga!** Dodawanie/usuwanie odbywa się przez trzykrotne kliknięcie przycisku w module Infibity podczas inclusion/exclusion. Poprawnie przeprowadzony proces zostanie potwierdzony pięciokrotnym mrugnięciem diody.

5.2. Obiekty

A. BINARY_SENSOR

Obiekt umożliwia odczyt stanu otwarcia/zamknięcia kontaktronu.

CECHY

Nazwa	Opis
`Value`	Zwraca stan wejścia: `0` - zamknięcie, `1` - otwarcie

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w przypadku zmiany stanu na przeciwny
`OnSwitchOn`	Zdarzenie wywoływane w momencie ustawienia stanu wysokiego na wejściu
`OnSwitchOff`	Zdarzenie wywoływane w momencie ustawienia stanu niskiego na wejściu

B. ZWAVE_BATTERY

Obiekt umożliwia odczyt stanu baterii. Odczyt odbywa się cyklicznie, co ustawiony czas, dla cechy `Interval` obiektu ZWAVE_WAKEUP.

CECHY

Nazwa	Opis
`BatteryLevel`	Poziom baterii modułu Z-Wave (w procentach)
`WarningLevel`	Poziom baterii, poniżej którego generowane są zdarzenia ostrzegawcze

METODY

Nazwa	Opis
`SetWarningLevel`	Ustawia poziom ostrzegawczy baterii modułu Z-Wave

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane przy zmianie wartości poziomu baterii
`OnLowBattery`	Zdarzenie wywoływane po wykryciu spadku poziomu baterii poniżej poziomu ostrzegawczego
`OnBatteryGood`	Zdarzenie wywoływane po wykryciu powrotu poziomu baterii do wartości powyżej poziomu ostrzegawczego

C. ZWAVE_WAKEUP

Obiekt umożliwia ustawienie oraz odczyt czasu wybudzenia bateryjnego modułu Z-Wave. Domyślna wartość ustawiana przez CLU wynosi 3600s (60 minut). Minimalna wartość to 300s (5 minut); maksymalna 16777200s (około 194 dni). Możliwe jest ustawienie wartości w kroku 60s (360s, 420s, 480s itd.)

CECHY

Nazwa	Opis
`Interval`	Okres samoczynnego wybudzania modułu Z-Wave z trybu uśpienia (w sekundach)
`LastWakeUp`	Czas ostatniego wybudzenia modułu Z-Wave z trybu uśpienia

METODY

Nazwa	Opis
`SetInterval`	Ustawia okres samoczynnego wybudzenia modułu Z-Wave z trybu uśpienia

ZDARZENIA

Nazwa	Opis
`OnWakeUp`	Zdarzenie wywoływane po wykryciu wybudzenia modułu Z-Wave z trybu uśpienia

D. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów oraz komunikacji z modułem w sieci Z-Wave. Umożliwia ustawienie zaawansowanych parametrów konfiguracyjnych danego modułu (określone indywidualnie w instrukcji).

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` – komunikacja z modułem nie jest zablokowana, `1` – zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie – jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)
`Register`	Numer rejestru (parametru) konfiguracyjnego, który ostatnio został odczytany/ustawiony za pomocą dostępnych metod
`Value`	Wartość rejestru (parametru) konfiguracyjnego **Uwaga!** *Parametr 1 i 2 odnosi się do asocjacji modułów, która nie jest wspierana przez system Grenton!*

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` *nie jest jednoznaczne z ponowną poprawną komunikacją z modułem - umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji
`Set`	Ustawia wartość danego rejestru (parametru) konfiguracyjnego: `Register` (numer rejestru lub parametru), `Value` (wartość rejestru bądź parametru), `Size` (rozmiar wysyłanej wartości rejestru lub parametru - w bajtach) **Uwaga!** *Wywołanie metody* `Set` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!*
`Get`	Pobiera wartość danego rejestru (parametru) konfiguracyjnego **Uwaga!** *Wywołanie metody* `Get` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!*
`SetDefault`	Ustawia wartość domyślną dla danego rejestru (parametru) konfiguracyjnego **Uwaga!** *Wywołanie metody* `SetDefault` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!*

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

6. INFIBITY Water Sensor [NEO Coolcam]

Wersja modułu: NAS-WS02ZU HW: 32 FW: 2.133

6.1. Informacje ogólne

Moduł Z-Wave Infibity Water Sensor umożliwia odczyt: stanu czujnika zalania oraz poziomu baterii. Ponadto daje możliwość ustawienia/odczytu czasu wybudzenia modułu.

> **Uwaga!** Dodawanie/usuwanie odbywa się przez trzykrotne kliknięcie przycisku w module Infibity podczas inclusion/exclusion. Poprawnie przeprowadzony proces zostanie potwierdzony pięciokrotnym mrugnięciem diody.

> **Uwaga!** Moduł w Object Managerze zgłasza się jako NEO COOLCAM!

6.2. Obiekty

A. BINARY_SENSOR

Obiekt umożliwia odczyt stanu czujnika zalania.

CECHY

Nazwa	Opis
`Value`	Zwraca stan wejścia: **0** - suchy, **1** - zalany

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w przypadku zmiany stanu na przeciwny
`OnSwitchOn`	Zdarzenie wywoływane w momencie ustawienia stanu wysokiego na wejściu
`OnSwitchOff`	Zdarzenie wywoływane w momencie ustawienia stanu niskiego na wejściu

B. ZWAVE_BATTERY

Obiekt umożliwia odczyt stanu baterii. Odczyt odbywa się cyklicznie, co ustawiony czas, dla cechy `Interval` obiektu `ZWAVE_WAKEUP`.

CECHY

Nazwa	Opis
`BatteryLevel`	Poziom baterii modułu Z-Wave (w procentach)
`WarningLevel`	Poziom baterii, poniżej którego generowane są zdarzenia ostrzegawcze

METODY

Nazwa	Opis
`SetWarningLevel`	Ustawia poziom ostrzegawczy baterii modułu Z-Wave

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane przy zmianie wartości poziomu baterii
`OnLowBattery`	Zdarzenie wywoływane po wykryciu spadku poziomu baterii poniżej poziomu ostrzegawczego
`OnBatteryGood`	Zdarzenie wywoływane po wykryciu powrotu poziomu baterii do wartości powyżej poziomu ostrzegawczego

C. ZWAVE_WAKEUP

Obiekt umożliwia ustawienie oraz odczyt czasu wybudzenia bateryjnego modułu Z-Wave. Domyślna wartość ustawiana przez CLU wynosi 3600s (60 minut). Minimalna wartość to 300s (5 minut); maksymalna 16777200s (około 194 dni). Możliwe jest ustawienie wartości w kroku 60s (360s, 420s, 480s itd.)

CECHY

Nazwa	Opis
`Interval`	Okres samoczynnego wybudzania modułu Z-Wave z trybu uśpienia (w sekundach)
`LastWakeUp`	Czas ostatniego wybudzenia modułu Z-Wave z trybu uśpienia

METODY

Nazwa	Opis
`SetInterval`	Ustawia okres samoczynnego wybudzenia modułu Z-Wave z trybu uśpienia

ZDARZENIA

Nazwa	Opis
`OnWakeUp`	Zdarzenie wywoływane po wykryciu wybudzenia modułu Z-Wave z trybu uśpienia

D. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów oraz komunikacji z modułem w sieci Z-Wave. Umożliwia ustawienie zaawansowanych parametrów konfiguracyjnych danego modułu (określone indywidualnie w instrukcji).

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` – komunikacja z modułem nie jest zablokowana, `1` – zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie – jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)
`Register`	Numer rejestru (parametru) konfiguracyjnego, który ostatnio został odczytany/ustawiony za pomocą dostępnych metod
`Value`	Wartość rejestru (parametru) konfiguracyjnego **Uwaga!** *Parametr 7 odnosi się do asocjacji modułów, która nie jest wspierana przez system Grenton!*

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` *nie jest jednoznaczne z ponowną poprawną komunikacją z modułem - umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji
`Set`	Ustawia wartość danego rejestru (parametru) konfiguracyjnego: `Register` (numer rejestru lub parametru), `Value` (wartość rejestru bądź parametru), `Size` (rozmiar wysyłanej wartości rejestru lub parametru - w bajtach) **Uwaga!** *Wywołanie metody* `Set` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!*
`Get`	Pobiera wartość danego rejestru (parametru) konfiguracyjnego **Uwaga!** *Wywołanie metody* `Get` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!*
`SetDefault`	Ustawia wartość domyślną dla danego rejestru (parametru) konfiguracyjnego **Uwaga!** *Wywołanie metody* `SetDefault` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!*

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

7. Heiman Smart Smoke Sensor

Wersja modułu: ***HS1SA-Z (HS1SA-Z HW: 255 FW: 1.10)***

7.1. Informacje ogólne

Moduł Z-Wave Heiman Smart Smoke Sensor umożliwia odczyt: stanu czujnika dymu oraz poziomu baterii. Ponadto daje możliwość ustawienia/odczytu czasu wybudzenia modułu.

> **Uwaga!** Dodawanie/usuwanie odbywa się przez trzykrotne kliknięcie przycisku w module HEIMAN podczas inclusion/exclusion. Poprawnie przeprowadzony proces zostanie potwierdzony pięciokrotnym mrugnięciem diody.

> **Uwaga!** Obsługa modułu dostępna na CLU z firmware 04.07.41 (Build 183201) i nowszym.

7.2. Obiekty

A. BINARY_SENSOR

Obiekt umożliwia odczyt stanu czujnika dymu.

CECHY

Nazwa	Opis
`Value`	Zwraca stan wejścia: **0** - brak naruszenia, **1** - naruszenie (dym)

METODY

-

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w przypadku zmiany stanu na przeciwny
`OnSwitchOn`	Zdarzenie wywoływane w momencie ustawienia stanu wysokiego na wejściu
`OnSwitchOff`	Zdarzenie wywoływane w momencie ustawienia stanu niskiego na wejściu

B. ZWAVE_BATTERY

Obiekt umożliwia odczyt stanu baterii. Odczyt odbywa się cyklicznie, co ustawiony czas, dla cechy `Interval` obiektu `ZWAVE_WAKEUP`.

CECHY

Nazwa	Opis
`BatteryLevel`	Poziom baterii modułu Z-Wave (w procentach)
`WarningLevel`	Poziom baterii, poniżej którego generowane są zdarzenia ostrzegawcze

METODY

Nazwa	Opis
`SetWarningLevel`	Ustawia poziom ostrzegawczy baterii modułu Z-Wave

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane przy zmianie wartości poziomu baterii
`OnLowBattery`	Zdarzenie wywoływane po wykryciu spadku poziomu baterii poniżej poziomu ostrzegawczego
`OnBatteryGood`	Zdarzenie wywoływane po wykryciu powrotu poziomu baterii do wartości powyżej poziomu ostrzegawczego

C. ZWAVE_WAKEUP

Obiekt umożliwia ustawienie oraz odczyt czasu wybudzenia bateryjnego modułu Z-Wave. Domyślna wartość ustawiana przez CLU wynosi 3600s (60 minut). Minimalna wartość to 300s (5 minut); maksymalna 16777200s (około 194 dni). Możliwe jest ustawienie wartości w kroku 60s (360s, 420s, 480s itd.)

CECHY

Nazwa	Opis
`Interval`	Okres samoczynnego wybudzania modułu Z-Wave z trybu uśpienia (w sekundach)
`LastWakeUp`	Czas ostatniego wybudzenia modułu Z-Wave z trybu uśpienia

METODY

Nazwa	Opis
`SetInterval`	Ustawia okres samoczynnego wybudzenia modułu Z-Wave z trybu uśpienia

ZDARZENIA

Nazwa	Opis
`OnWakeUp`	Zdarzenie wywoływane po wykryciu wybudzenia modułu Z-Wave z trybu uśpienia

D. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów komunikacji z modułem w sieci Z-Wave.

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` – komunikacja z modułem nie jest zablokowana, `1` – zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie – jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** *`RemoveBan` nie jest jednoznaczne z ponowną poprawną komunikacją z modułem – umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

8. INFIBITY Siren Alarm [NEO Coolcam]

Wersja modułu: ***NAS-AB01Z HW:48 FW: 2.90***

8.1. Informacje ogólne

Obsługa modułu Infibity Siren Alarm obejmuje możliwość załączania / wyłączania sygnału syreny, odczyt poziomu baterii, a także ustawienie oraz odczyt wybudzenia modułu. Dodatkowo istnieje możliwość zmiany parametrów konfiguracyjnych.

Sposób dodawania / usuwania: W celu dodania / usunięcia modułu należy 3x kliknąć na przycisku w module Infibity podczas inclusion / exclusion - w przypadku poprawnego dodania / usunięcia, czerwone LEDy zamrugają 5 razy.

> **Uwaga!** Po restarcie (wysłaniu konfiguracji) CLU, należy odczekać 10s przed pierwszą próbą załączenia modułu Siren Alarm.

8.2. Obiekty

A. ZWAVE_DOUT

Obiekt umożliwia załączenie / wyłączenie oraz odczyt aktualnego stanu syreny.

CECHY

Nazwa	Opis
`Value`	Zwraca stan wyjścia (0 - wyłączone; 1 - załączone)

METODY

Nazwa	Opis
`SetValue`	Ustawia stan wyjścia jako 1 lub 0
`Switch`	Przełącza wyjście. Parametr Time określa na jak długo następuje zmiana stanu, dla 0 jest ona stała
`SwitchOn`	Załącza wyjście. Parametr Time określa na jak długo następuje zmiana stanu, dla 0 jest ona stała
`SwitchOff`	Wyłącza wyjście. Parametr Time określa na jak długo następuje zmiana stanu, dla 0 jest ona stała

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w przypadku zmiany stanu na przeciwny
`OnSwitchOn`	Zdarzenie wywoływane w momencie ustawienia stanu wysokiego na wyjściu
`OnSwitchOff`	Zdarzenie wywoływane w momencie ustawienia stanu niskiego na wyjściu

B. ZWAVE_BATTERY

Obiekt umożliwia odczyt stanu baterii. Odczyt stanu odbywa się cyklicznie co ustawiony czas dla cechy Interval obiektu ZWAVE_WAKEUP

CECHY

Nazwa	Opis
`BatteryLevel`	Poziom baterii modułu Z-Wave w procentach
`WarningLevel`	Poziom baterii, poniżej którego generowane są zdarzenia ostrzegawcze

METODY

Nazwa	Opis
`SetWarningLevel`	Ustawia poziom ostrzegawczy baterii modułu Z-Wave

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane
`OnLowBattery`	Zdarzenie wywoływane po wykryciu spadku poziomu baterii poniżej poziomu ostrzegawczego
`OnBatteryGood`	Zdarzenie wywoływane po wykryciu powrotu poziomu baterii do wartości powyżej poziomu ostrzegawczego

C. ZWAVE_WAKEUP

Obiekt umożliwiający ustawienie oraz odczyt czasu wybudzenia bateryjnego modułu Z-Wave. Domyślna wartość ustawiania przez CLU wynosi 3600s (5 minut). Minimalna wartość to 60s (1 minuta); maksymalna 16777200s (około 194 dni).

CECHY

Nazwa	Opis
`Interval`	Okres samoczynnego wybudzania modułu Z-Wave z trybu uśpienia w sekundach
`LastWakeUp`	Czas ostatniego wybudzenia modułu Z-Wave z trybu uśpienia

METODY

Nazwa	Opis
`SetInterval`	Ustawia okres samoczynnego wybudzania modułu Z-Wave z trybu uśpienia

ZDARZENIA

Nazwa	Opis
`OnWakeUp`	Zdarzenie wywoływane po wykryciu wybudzenia modułu Z-Wave z trybu uśpienia

D. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów komunikacji z modułem w sieci Z-Wave. Umożliwia również ustawienie zaawansowanych parametrów konfiguracyjnych danego modułu (określone indywidualnie w instrukcji).

Ustawienie rejestru 7 zmienia tryb pracy syreny:

- Jako **Alarm** - syrena działa zgodnie z ustawieniami parametrów: 1,2,5,8
- Jako **DoorBell** - syrena działa zgodnie z ustawieniami parametrów: 3,4,6,9

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` - komunikacja z modułem nie jest zablokowana, `1` - zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie - jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1). **Uwaga!** Po restarcie CLU, do modułu wysyłana jest komenda *Switch Binary Switch* na którą moduł nie odpowiada, przez co `FailCount` zostaje zwiększony o 1.
`Register`	Numer rejestru (parametru) konfiguracyjnego, który ostatnio został odczytany/ustawiony za pomocą dostępnych metod
`Value`	Wartość rejestru (parametru) konfiguracyjnego

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` nie jest jednoznaczne z ponowną poprawną komunikacją z modułem – umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji
`Set`	Ustawia wartość danego rejestru (parametru) konfiguracyjnego: `Register` (numer rejestru lub parametru), `Value` (wartość rejestru bądź parametru), `Size` (rozmiar wysyłanej wartości rejestru lub parametru – w bajtach) **Uwaga!** *Wywołanie metody* `Set` *musi być wykonane po wybudzeniu modułu bateryjnego! W celu wybudzenia modułu należy trzykrotnie kliknąć przycisk w module - po wybudzeniu zamruga czerwona dioda!*
`Get`	Pobiera wartość danego rejestru (parametru) konfiguracyjnego
`SetDefault`	Ustawia wartość domyślną dla danego rejestru (parametru) konfiguracyjnego

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

9. Danfoss Living Connect

Wersja modułu: EU HW: 00 FW: 1.1

9.1. Informacje ogólne

Obsługa modułu Danfoss Living Connect obejmuje możliwość ustawiania temperatury zadanej na głowicy, a także załączania / wyłączania blokady przycisków. Możliwy jest również odczyt poziomu naładowania baterii urządzenia oraz definiowanie okresu wybudzania modułu.

Sposób dodawania / usuwania: Aby dodać / usunąć urządzenie należy 1x kliknąć środkowy przycisk na module podczas inclusion / exclusion (wywołane na CLU) – podświetlenie wyświetlacza zacznie szybko mrugać, a następnie zaświeci światłem ciągłym. Jeśli po dłuższym czasie szybkiego mrugania podświetlenie wyświetlacza zacznie mrugać wolniej, to oznacza, że proces dodawania się nie udał. Przed dodaniem urządzenia należy wyjść z trybu montażu sygnalizowanego na wyświetlaczu symbolem „M”.

9.2. Obiekty

A. ZWAVE_THERMOSTAT

Obiekt umożliwiający ustawianie temperatury zadanej na głowicy a także załączania / wyłączania blokady przycisków.

> **Uwaga!** Obsługa nie obejmuje odczytu ustawionej temperatury za pomocą przycisków na głowicy.

CECHY

Nazwa	Opis
`PointValue`	Zwraca wartość zadanej temperatury (4°C ÷ 28°C)
`ProtectionState`	Zwraca stan blokady klawiszy: `0` - wyłączona, `2` - włączona

METODY

Nazwa	Opis
`SetPointValue`	Ustawia wartość zadanej temperatury (cecha PointValue)
`SetProtectionState`	Ustawia stan blokady klawiszy

ZDARZENIA

Nazwa	Opis
`OnPointValueChange`	Zdarzenie wywoływane w przypadku zmiany wartości zadanej temperatury
`OnProtectionChange`	Zdarzenie wywoływane w momencie zmiany stanu blokady klawiszy
`OnProtectionOn`	Zdarzenie wywoływane w momencie włączenia blokady klawiszy
`OnProtectionOff`	Zdarzenie wywoływane w momencie wyłączenia blokady klawiszy

B. ZWAVE_BATTERY

Obiekt umożliwia odczyt stanu baterii. Odczyt stanu odbywa się cyklicznie co ustawiony czas dla cechy Interval obiektu ZWAVE_WAKEUP

CECHY

Nazwa	Opis
`BatteryLevel`	Poziom baterii modułu Z-Wave w procentach
`WarningLevel`	Poziom baterii, poniżej którego generowane są zdarzenia ostrzegawcze

METODY

Nazwa	Opis
`SetWarningLevel`	Ustawia poziom ostrzegawczy baterii modułu Z-Wave

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane
`OnLowBattery`	Zdarzenie wywoływane po wykryciu spadku poziomu baterii poniżej poziomu ostrzegawczego
`OnBatteryGood`	Zdarzenie wywoływane po wykryciu powrotu poziomu baterii do wartości powyżej poziomu ostrzegawczego

C. ZWAVE_WAKEUP

Obiekt umożliwiający ustawienie oraz odczyt czasu wybudzenia bateryjnego modułu Z-Wave. Domyślna wartość ustawiania przez CLU wynosi 300s (5 minut). Minimalna wartość to 60s (1 minuta); maksymalna 1800s (30 minut). Możliwe jest ustawienie wartości w kroku 60s (60s, 120s, 180s itd.)

CECHY

Nazwa	Opis
`Interval`	Okres samoczynnego wybudzania modułu Z-Wave z trybu uśpienia w sekundach
`LastWakeUp`	Czas ostatniego wybudzenia modułu Z-Wave z trybu uśpienia

METODY

Nazwa	Opis
`SetInterval`	Ustawia okres samoczynnego wybudzania modułu Z-Wave z trybu uśpienia

ZDARZENIA

Nazwa	Opis
`OnWakeUp`	Zdarzenie wywoływane po wykryciu wybudzenia modułu Z-Wave z trybu uśpienia

D. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów komunikacji z modułem w sieci Z-Wave.

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` - komunikacja z modułem nie jest zablokowana, `1` - zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie - jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` *nie jest jednoznaczne z ponowną poprawną komunikacją z modułem - umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

10. POPP Z-Weather

Wersja modułu: EU HW: 01 FW: 1.0

10.1. Informacje ogólne

Obsługa modułu POPP Z-Weather obejmuje możliwość odczytu parametrów klimatycznych ze stacji pogodowej. Możliwy jest również odczyt poziomu naładowania baterii urządzenia, a także definiowanie okresu wybudzania modułu.

Sposób dodawania / usuwania: Aby dodać / usunąć urządzenie należy 3x kliknąć przycisk na module w czasie 1,5s podczas inclusion / exclusion (wywołane na CLU) – czerwona dioda na module zamruga 3x podczas dodawania lub 1x podczas usuwania.

Sposób wybudzenia urządzenia: 1x kliknąć na urządzeniu

10.2. Obiekty

A. ZWAVE_WEATHER

Obiekt umożliwiający odczyt parametrów klimatycznych - temperatury, luminacji, wilgotności względnej, prędkości wiatru, ciśnienia barometrycznego oraz temperatury punktu rosy.

CECHY

Nazwa	Opis
`Temperature`	Zwraca wartość zmierzonej temperatury powietrza (-10°C ÷ 60°C)
`Luminance`	Zwraca wartość zmierzonej luminancji (0% ÷ 100%)
`Humidity`	Zwraca wartość zmierzonej wilgotności względnej (0% ÷ 100%)
`WindSpeed`	Zwraca wartość zmierzonej prędkości wiatru (0m/s ÷ 31m/s)
`Pressure`	Zwraca wartość zmierzonego ciśnienia barometrycznego (600hPa ÷ 1200hPa)
`DewPoint`	Zwraca wartość zmierzonej temperatury punktu rosy (-56,4°C ÷ 60°C)

ZDARZENIA

Nazwa	Opis
`OnTemperatureChange`	Zdarzenie wywoływane w przypadku zmiany wartości temperatury powietrza
`OnLuminanceChange`	Zdarzenie wywoływane w momencie zmiany wartości luminancji
`OnHumidityChange`	Zdarzenie wywoływane w momencie zmiany wartości wilgotności względnej
`OnWindSpeedChange`	Zdarzenie wywoływane w momencie zmiany wartości prędkości wiatru
`OnPressureChange`	Zdarzenie wywoływane w momencie zmiany wartości ciśnienia barometrycznego
`OnDewPointChange`	Zdarzenie wywoływane w momencie zmiany wartości temperatury punktu rosy

B. ZWAVE_BATTERY

Obiekt umożliwia odczyt stanu baterii. Odczyt stanu odbywa się cyklicznie co ustawiony czas dla cechy Interval obiektu ZWAVE_WAKEUP

CECHY

Nazwa	Opis
`BatteryLevel`	Poziom baterii modułu Z-Wave w procentach
`WarningLevel`	Poziom baterii, poniżej którego generowane są zdarzenia ostrzegawcze

METODY

Nazwa	Opis
`SetWarningLevel`	Ustawia poziom ostrzegawczy baterii modułu Z-Wave

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane
`OnLowBattery`	Zdarzenie wywoływane po wykryciu spadku poziomu baterii poniżej poziomu ostrzegawczego
`OnBatteryGood`	Zdarzenie wywoływane po wykryciu powrotu poziomu baterii do wartości powyżej poziomu ostrzegawczego

C. ZWAVE_WAKEUP

Obiekt umożliwiający ustawienie oraz odczyt czasu wybudzenia bateryjnego modułu Z-Wave. Domyślna wartość ustawiania przez CLU wynosi 600s (około 10 minut). Minimalna wartość to 600s (około 10 minut), maksymalna 17180s (około 286 minut).Możliwe jest ustawienie wartości w kroku 1s (600s, 601s, 602s itd.)

CECHY

Nazwa	Opis
`Interval`	Okres samoczynnego wybudzania modułu Z-Wave z trybu uśpienia w sekundach
`LastWakeUp`	Czas ostatniego wybudzenia modułu Z-Wave z trybu uśpienia

METODY

Nazwa	Opis
`SetInterval`	Ustawia okres samoczynnego wybudzania modułu Z-Wave z trybu uśpienia

ZDARZENIA

Nazwa	Opis
`OnWakeUp`	Zdarzenie wywoływane po wykryciu wybudzenia modułu Z-Wave z trybu uśpienia

D. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów komunikacji z modułem w sieci Z-Wave.

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` - komunikacja z modułem nie jest zablokowana, `1` - zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie - jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` *nie jest jednoznaczne z ponowną poprawną komunikacją z modułem - umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

11. FAKRO AMZ Solar

Wersja modułu: HW: 31 FW: 1.01:01.01

11.1. Informacje ogólne

Obsługa modułu FAKRO AMZ Solar obejmuje możliwość sterowania oknem - zarówno poprzez maksymalne otwarcie / zamknięcie, jak i ustawienie wartości procentowej otwarcia okna, zmianę trybu pracy (również trybu sezonowego), a także definiowanie parametrów działających w danym trybie. Ponadto daje możliwość zmiany parametrów konfiguracyjnych (interfejs konfiguracyjny Fakro).

Sposób dodawania / usuwania: Dodawanie / usuwanie urządzenia odbywa się poprzez naciśnięcie przycisku 'P' na urządzeniu podczas inclusion / exclusion (wywołane na CLU).

11.2. Obiekty

ZWAVE_FAKRO

Obiekt umożliwiający sterowanie otwarciem markizy i odczyt ustawionego procentu otwarcia. Dostępne jest ustawianie na wartość maksymalną (otwarcie / zamknięcie), a także podając procentową wartość otwarcia markizy (0-100%). Dodatkowo możliwe jest ustawianie trybów pracy urządzenia oraz parametrów dotyczących poszczególnych trybów pracy.

> **Uwaga!** Informacje dotyczące poszczególnych trybów pracy znajdują się w dokumentacji urządzenia dostarczanej przez producenta.

CECHY

Nazwa	Opis
`State`	Stan urządzenia: `0` - stoi, `1` - ruch w górę, `2` - ruch w dół
`Percent`	Wartość procentowa otwarcia markizy, gdzie: 0% - okno zamknięte, 100% - okno otwarte **Uwaga!** *Wartość cechy* `Percent` *jest odświeżana w momencie, gdy sterownik markizy zakończy daną pracę - należy mieć to na uwadze przy wykorzystywaniu tej cechy np. dla komponentu Slider.*
`Mode`	Tryb pracy urządzenia: `0 - Manual` - Ręczny, `1 - Semiauto` - Półautomatyczny, `2 - Auto` - Automatyczny
`SeasonMode`	Tryb sezonowy urządzenia: `0 - Summer` - Lato, `1 - Winter` - Zima **Uwaga!** *Parametr nie dotyczy trybu ręcznego* `Mode = 0`
`OpeningTime`	Czas otwarcia markizy w trybie półautomatycznym
`Sensitivity`	Czułość poziomu nasłonecznienia dla markizy w trybie automatycznym

Uwaga! Wartość ustawionych parametrów konfiguracyjnych jest odświeżana w momencie `WakeUp`'u danego urządzenia (pobierane są wartości od urządzenia Z-Wave).

METODY

Nazwa	Opis
`Up`	Markiza do góry
`Down`	Markiza w dół
`Stop`	Stop, jeśli markiza jest w ruchu
`Start`	Markiza do góry, jeśli poprzednio ruch w dół, markiza w dół, jeśli poprzednio ruch w górę
`SetPercent`	Ustawia wartość procentową, gdzie 100% - markiza otwarta
`SetMode`	Ustawia tryb pracy urządzenia
`SetSeasonMode`	Ustawia tryb sezonowy
`SetOpeningTime`	Ustawia czas otwarcia markizy
`SetSensitivity`	Ustawia czułość poziomu nasłonecznienia

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w momencie zmiany stanu sterownika okna
`OnUp`	Zdarzenie wywoływane w momencie zmiany stanu ze Stop na Up
`OnDown`	Zdarzenie wywoływane w momencie zmiany stanu ze Stop na Down
`OnStart`	Zdarzenie wywoływane w momencie wywołania komendy Start
`OnStop`	Zdarzenie wywoływane w momencie wywołania komendy Stop

ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów komunikacji z modułem w sieci Z-Wave. Umożliwia ustawienie zaawansowanych parametrów konfiguracyjnych danego modułu (określone indywidualnie w instrukcji).

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` - komunikacja z modułem nie jest zablokowana, `1` - zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie - jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)
`Register`	Numer rejestru (parametru) konfiguracyjnego, który ostatnio został odczytany/ustawiony za pomocą dostępnych metod
`Value`	Wartość rejestru (parametru) konfiguracyjnego

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` *nie jest jednoznaczne z ponowną poprawną komunikacją z modułem - umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji
`Set`	Ustawia wartość danego rejestru (parametru) konfiguracyjnego: `Register` (numer rejestru lub parametru), `Value` (wartość rejestru bądź parametru), `Size` (rozmiar wysyłanej wartości rejestru lub parametru - w bajtach)
`Get`	Pobiera wartość danego rejestru (parametru) konfiguracyjnego
`SetDefault`	Ustawia wartość domyślną dla danego rejestru (parametru) konfiguracyjnego

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

12. FAKRO ARF

12.1. Informacje ogólne

Obsługa modułu FAKRO ARF obejmuje możliwość sterowania roletą - zarówno maksymalne otwarcie / zamknięcie, jak i ustawienie wartości procentowej otwarcia rolety.

Sposób dodawania / usuwania: Dodawanie / usuwanie urządzenia odbywa się poprzez naciśnięcie przycisku 'P' na urządzeniu podczas inclusion / exclusion (wywołane na CLU).

12.2. Obiekty

A. ZWAVE_FAKRO

Obiekt umożliwiający sterowanie roletą i odczyt ustawionego procentu otwarcia. Dostępne jest ustawianie na wartość maksymalną (otwarcie / zamknięcie) a także podając procentową wartość otwarcia rolety (0-100%).

CECHY

Nazwa	Opis
`State`	Stan rolety: `0` - stoi, `1` - ruch w górę, `2` - ruch w dół
`Percent`	Wartość procentowa otwarcia rolety, gdzie: `0%` - roleta zamknięta, `100%` - roleta otwarta **Uwaga!** Wartość cechy `Percent` jest odświeżana w momencie, gdy roleta zakończy daną pracę - należy mieć to na uwadze przy wykorzystywaniu tej cech np. dla komponentu Slider. **Uwaga!** *Wywołanie metody `Stop` podczas jazdy rolety nie powoduje odświeżenia cechy `Percent`*

METODY

Nazwa	Opis
`Up`	Roleta do góry
`Down`	Roleta w dół
`Stop`	Stop, jeśli roleta jest w ruchu
`Start`	Roleta do góry, jeśli poprzednio ruch w dół, roleta w dół, jeśli poprzednio ruch w górę
`SetPercent`	Ustawia wartość procentową, gdzie 100% - roleta otwarta

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w momencie zmiany stanu rolety
`OnUp`	Zdarzenie wywoływane w momencie zmiany stanu ze Stop na Up
`OnDown`	Zdarzenie wywoływane w momencie zmiany stanu ze Stop na Down
`OnStart`	Zdarzenie wywoływane w momencie wywołania komendy Start
`OnStop`	Zdarzenie wywoływane w momencie wywołania komendy Stop

B. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów komunikacji z modułem w sieci Z-Wave.

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` – komunikacja z modułem nie jest zablokowana, `1` – zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie – jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` *nie jest jednoznaczne z ponowną poprawną komunikacją z modułem – umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

13. FAKRO FTP_V

Wersja modułu: HW: 25 FW: 1.01:01.01

13.1. Informacje ogólne

Obsługa modułu FAKRO FTP_V obejmuje możliwość sterowania oknem - zarówno poprzez maksymalne otwarcie / zamknięcie, jak i ustawienie wartości procentowej otwarcia okna.

Sposób dodawania / usuwania: Dodawanie / usuwanie urządzenia odbywa się poprzez naciśnięcie przycisku 'P' na urządzeniu podczas inclusion / exclusion (wywołane na CLU).

13.2. Obiekty

A. ZWAVE_FAKRO

Obiekt umożliwiający sterowanie otwarciem oknem i odczyt ustawionego procentu otwarcia. Dostępne jest ustawianie na wartość maksymalną (otwarcie / zamknięcie), a także podając procentową wartość otwarcia okna (0-100%).

CECHY

Nazwa	Opis
`State`	Stan urządzenia: `0` - stoi, `1` - otwieranie, `2` - zamykanie
`Percent`	Wartość procentowa otwarcia okna, gdzie: `0%` - okno zamknięte, `100%` - okno otwarte **Uwaga!** *Wartość cechy `Percent` jest odświeżana w momencie, gdy sterownik okna zakończy daną pracę - należy mieć to na uwadze przy wykorzystywaniu tej cech np. dla komponentu Slider.*
WaterSensor	Wartość z czujnika deszczu

METODY

Nazwa	Opis
`Open`	Otwarcie okna
`Close`	Zamknięcie okna
`Stop`	Stop, jeśli okno jest otwierane lub zamykane
`Start`	Zamknięcie okna, jeśli poprzednio było otwierane, otwieranie okna, jeśli poprzednio było zamykane
`SetPercent`	Ustawia wartość procentową, gdzie `100%` - okno otwarte

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w momencie zmiany stanu sterownika okna
`OnOpen`	Zdarzenie wywoływane w momencie zmiany stanu ze Stop na Open
`OnClose`	Zdarzenie wywoływane w momencie zmiany stanu ze Stop na Close
`OnStart`	Zdarzenie wywoływane w momencie wywołania komendy Start
`OnStop`	Zdarzenie wywoływane w momencie wywołania komendy Stop
`OnRainChange`	Zdarzenie wywoływane w przypadku zmiany stanu czujnika na przeciwny
`OnRainOn`	Zdarzenie wywoływane w momencie ustawienia stanu wysokiego na czujniku
`OnRainOff`	Zdarzenie wywoływane w momencie ustawienia stanu niskiego na czujniku

B. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów komunikacji z modułem w sieci Z-Wave.

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` - komunikacja z modułem nie jest zablokowana, `1` - zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie - jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)
`Register`	Numer rejestru (parametru) konfiguracyjnego, który ostatnio został odczytany/ustawiony za pomocą dostępnych metod
`Value`	Wartość rejestru (parametru) konfiguracyjnego

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` *nie jest jednoznaczne z ponowną poprawną komunikacją z modułem - umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji
`Set`	Ustawia wartość danego rejestru (parametru) konfiguracyjnego: `Register` (numer rejestru lub parametru), `Value` (wartość rejestru bądź parametru), `Size` (rozmiar wysyłanej wartości rejestru lub parametru - w bajtach)
`Get`	Pobiera wartość danego rejestru (parametru) konfiguracyjnego
`SetDefault`	Ustawia wartość domyślną dla danego rejestru (parametru) konfiguracyjnego

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

14. Fibaro RGBW

Wersja modułu: *FGRGBWM-441 v2/5 EU*

14.1. Informacje ogólne

Moduł Z-Wave Fibaro RGBW umożliwia odczytywanie oraz ustawianie stanu pojedynczych kanałów wyjściowych R, G, B, W w zakresie od 0 do 255. Ponadto daje możliwość zmiany parametrów konfiguracyjnych (interfejs konfiguracyjny Fibaro).

14.2. Obiekty

A. ZWAVE_RGBW_LED

Obiekt umożliwia ustawianie wartości (0-255) dla pojedynczych kanałów wyjściowych R, G, B, W. Możliwy jest również odczyt tych wartości - np. ustawione bezpośrednio z przycisku podłączonego do modułu.

> **Uwaga!** Wartość z załączonego przycisku jest wysyłana w momencie zwolnienia lub doprowadzenia do wartości minimalnej/maksymalnej!

CECHY

Nazwa	Opis
`Red`	Wartość składowej R (0-255) - kolor czerwony
`Green`	Wartość składowej G (0-255) - kolor zielony
`Blue`	Wartość składowej B (0-255) - kolor niebieski
`White`	Wartość składowej W (0-255) - kolor biały
`RampTime`	Czas narastania/opadania zmiany wartości ściemniacza w milisekundach. Wartość tej cechy wpływa na akcje wywoływane przez CLU – nie wpływa na czas narastania/opadania po naciśnięciu przycisków podłączonych bezpośrednio do modułu

METODY

Nazwa	Opis
`SetRed`	Ustawia wartość składowej R (0-255) - kolor czerwony
`SetGreen`	Ustawia wartość składowej G (0-255) - kolor zielony
`SetBlue`	Ustawia wartość składowej B (0-255) - kolor niebieski
`SetWhite`	Ustawia wartość składowej W (0-255) - kolor biały
`SetRampTime`	Ustawia czas narastania/opadania zmiany wartości ściemniacza

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane przy zmianie wartości ściemniacza
`OnSwitchOn`	Zdarzenie wywoływane przy zmianie stanu ściemniacza na włączony
`OnSwitchOff`	Zdarzenie wywoływane przy zmianie stanu ściemniacza na wyłączony

B. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów oraz komunikacji z modułem w sieci Z-Wave. Umożliwia ustawienie zaawansowanych parametrów konfiguracyjnych danego modułu (określone indywidualnie w instrukcji).

> **Uwaga!** W przypadku modułów Fibaro RGBW dodanych już do projektu - obiekt ZWAVE_CONFIG zostanie dodany tylko w momencie całkowitego usunięcia modułu z projektu i po wykonaniu CLU Discovery.

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` – komunikacja z modułem nie jest zablokowana, `1` – zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie – jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)
`Register`	Numer rejestru (parametru) konfiguracyjnego, który ostatnio został odczytany/ustawiony za pomocą dostępnych metod
`Value`	Wartość rejestru (parametru) konfiguracyjnego

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** *Cecha `RemoveBan` nie jest jednoznaczne z ponowną poprawną komunikacją z modułem – umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji
`Set`	Ustawia wartość danego rejestru (parametru) konfiguracyjnego: `1 - Register` (numer rejestru lub parametru), `2 - Value` (wartość rejestru bądź parametru), `3 - Size` (rozmiar wysyłanej wartości rejestru lub parametru – w bajtach)
`Get`	Pobiera wartość danego rejestru (parametru) konfiguracyjnego
`SetDefault`	Ustawia wartość domyślną dla danego rejestru (parametru) konfiguracyjnego

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

15. Remotec ZXT-120

Wersja modułu: ZXT-120EU V1.0

15.1. Informacje ogólne

Obsługa modułu Remotec ZXT-120 obejmuje możliwość nauki a także wysyłania kodu IR, definiowanie parametrów transmisji oraz odczyt stanu uczenia danego kodu przez urządzenie. Możliwe jest również definiowanie okresu wybudzania modułu.

Sposób dodawania / usuwania: 1x kliknąć przycisk *PROG* w module podczas inclusion / exclusion - czerwona dioda zamruga 1x, a następnie zacznie świecić światłem ciągłym.

Sposób przywracania urządzenia do ustawień fabrycznych: należy przytrzymać przycisk *PROG* na urządzeniu przez 10 sekund. Po ok. 5 sekundach dioda czerwona zaświeci się a następnie zacznie mrugać dwukrotnie w momencie zakończenia procesu (ok. 10 sekund).

15.2. Opis konfiguracji urządzenia

Urządzenie można skonfigurować na dwa sposoby:

1. Nauczenie własnych kodów IR
2. Wykorzystanie z listy pre-definiowanych kodów dostępnych w wewnętrznej biblioteki kodów IR

A. Sposób uczenia kodów IR

1. Uczenie kodów odbywa się za pomocą głównego obiektu ZWAVE_IR1
2. Wywołać metodę `SetAcDeviceNumber` z parametrem `AcDeviceNumber` równym "0000" - ustawia urządzenie w tryb uczenia nowych kodów (spoza listy pre-definiowanej). Po wywołaniu metody, na module dioda LED zamruga 2x.
3. Wywołać metodę `LearnCode` podając numer kodu IR z przedziału 0-22 pod jakim chcemy, aby kod się zapisał. Po wywołaniu metody, dioda na urządzeniu powinna zgasnąć i zaświecić się ponownie.
4. W ciągu 15 sekund nacisnąć i trzymać przycisk pilota, którego chcemy nauczyć kierując pilot w kierunku górnej części urządzenia w odległości 1-3 cm.
    - W przypadku poprawnego zaprogramowania kodu IR dioda na urządzeniu powinna mrugnąć 2x.
    - W przypadku niepowodzenia dioda na urządzeniu powinna mrugnąć 6x.

Status uczenia można także odczytać z parametru `LearningStatus`. Dodatkowo generowane są odpowiednie zdarzenia w zależności od statusu uczenia (`OnLearning`, `OnLearningOK`, `OnLearningFail`)

> **Uwaga!** Pozycja pilota względem urządzenia w czasie uczenia jest kluczowa. Zaleca się, aby pilot był nieruchomy względem urządzenia podczas naciskania przycisku. Niepoprawna pozycja może spowodować, że zapamiętany kod będzie błędny mimo poprawnego statusu uczenia.

Uwaga! Pamięć nauczonych kodów jest zachowywana po odłączeniu zasilania urządzenia. Pamięć ta jest czyszczona po zmianie numeru urządzenia AC oraz po usunięciu urządzenia z sieci Z-Wave.

B. Sposób wysyłania kodów IR

1. Wywołać komendę `SendCode` podając numer nauczonego kodu IR z przedziału 0-22.
2. Po wywołaniu metody dioda na urządzeniu powinna zgasnąć i zaświecić się ponownie a przypisany kod jest wysłany do urządzenia docelowego.

Uwaga! Zewnętrzny nadajnik ma bardzo małą moc i mały kąt świecenia, dlatego powinny być umieszczone blisko odbiornika IR kontrolowanego urządzenia i odpowiednio skierowane. Kierunek świecenia nadajników IR jest zgodny z osią przewodu wchodzącego do obudowy nadajnika IR.

Uwaga! Zaleca się nie zmieniać numeru urządzenia AC (cecha AcDeviceNumber), jeśli nie korzystamy z wewnętrznej bazy kodów IR urządzenia.

15.3. Obiekty

A. ZWAVE_IR

Obiekt umożliwia odczyt i zapis parametrów konfiguracyjnych oraz wysyłanie kodów IR.

CECHY

Nazwa	Opis
`AcDeviceNumber`	Zwraca numer urządzenia AC z wewnętrznej biblioteki kodów IR (numer z Listy Kodów ZXT-120)
`EmitterPower`	Zwraca moc zewnętrznego (podłączonego) nadajnika podczerwieni: `0` – normalna moc `255` – duża moc
`LearningStatus`	Zwraca status trybu uczenia kodów IR: `0` – kanał IR bezczynny, `1` – uczenie zakończone powodzeniem, `2` – trwa procedura uczenia, `4` – uczenie zakończone niepowodzeniem
`SurroundIrControl`	Wielokierunkowa transmisja sygnału IR: `0` - Disable (wyłączona), `255` - Enable (załączona)

Uwaga! Wartość ustawionych parametrów konfiguracyjnych jest odświeżana w momencie `WakeUp`'u danego urządzenia (pobierane są wartości od urządzenia Z-Wave). Na czas konfigurowania parametrów urządzenia (`SetAcDeviceNumber`, `SetEmitterPower`, `SetSurroundIrControl`) oraz poprawnego odczytu ustawionych cech, możliwe jest ustawienie czasu `` `WakeUpInterval `` na czas mniejszy niż 60s. Po dokonaniu zmian i zakończeniu konfiguracji powyższych parametrów należy zmienić czas wybudzania na wartość co najmniej 60s.

METODY

Nazwa	Opis
`SendCode`	Wysyła kod IR o określonym numerze (numer kodu z zakresu 0-22, nauczony lub dostępny w wewnętrznej bibliotece kodów IR dla danego urządzenia AC)
`LearnCode`	Wywołuje tryb uczenia kodu IR o określonym numerze (numer kodu z zakresu 0-22)
`SetAcDeviceNumber`	Ustawia numer urządzenia AC z wewnętrznej biblioteki kodów IR (numer z Listy Kodów ZXT-120)
`SetEmitterPower`	Ustawia moc zewnętrznego nadajnika podczerwieni
`SetSurroundIrControl`	Ustawia wielokierunkowość sygnału IR

ZDARZENIA

Nazwa	Opis
`OnIrSend`	Zdarzenie wywoływane w momencie wysłania kodu IR
`OnLearningStatusChange`	Zdarzenie wywoływane w momencie zmiany statusu trybu uczenia kodu IR
`OnLearningOK`	Zdarzenie wywoływane w momencie zmiany statusu trybu uczenia kodu IR na „OK”
`OnLearning`	Zdarzenie wywoływane w momencie zmiany statusu trybu uczenia kodu IR na „Learning”
`OnLearningFail`	Zdarzenie wywoływane w momencie zmiany statusu trybu uczenia kodu IR na „Learning Fail”

B. ZWAVE_BATTERY

Obiekt umożliwia odczyt stanu baterii. Odczyt stanu odbywa się cyklicznie co ustawiony czas dla cechy Interval obiektu ZWAVE_WAKEUP

CECHY

Nazwa	Opis
`BatteryLevel`	Poziom baterii modułu Z-Wave w procentach
`WarningLevel`	Poziom baterii, poniżej którego generowane są zdarzenia ostrzegawcze

METODY

Nazwa	Opis
`SetWarningLevel`	Ustawia poziom ostrzegawczy baterii modułu Z-Wave

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane
`OnLowBattery`	Zdarzenie wywoływane po wykryciu spadku poziomu baterii poniżej poziomu ostrzegawczego
`OnBatteryGood`	Zdarzenie wywoływane po wykryciu powrotu poziomu baterii do wartości powyżej poziomu ostrzegawczego

C. ZWAVE_WAKEUP

Obiekt umożliwiający ustawienie oraz odczyt czasu odczytu parametrów modułu Z-Wave. Domyślna wartość ustawiania przez CLU wynosi 3600s (60 minut). Minimalna wartość to 10s, maksymalna 16777200s (około 194 dni). Możliwe jest ustawienie wartości w kroku 5s.

> **Uwaga!** Niezalecane jest ustawianie wartości cechy `WakeUp` mniejszej niż 60s podczas normalnej pracy urządzenia. Zmniejszenie wartości może być przydatne tylko w przypadku 'nauczania' kodów przez urządzenie (generowanie zdarzeń zmiany statusu trybu uczenia, a także odczyt cechy `LearningStatus`), jak również przy ustawianiu parametrów konfiguracyjnych.

CECHY

Nazwa	Opis
`Interval`	Okres samoczynnego wybudzania modułu Z-Wave z trybu uśpienia w sekundach
`LastWakeUp`	Czas ostatniego wybudzenia modułu Z-Wave z trybu uśpienia

METODY

Nazwa	Opis
`SetInterval`	Ustawia okres samoczynnego wybudzania modułu Z-Wave z trybu uśpienia

ZDARZENIA

Nazwa	Opis
`OnWakeUp`	Zdarzenie wywoływane po wykryciu wybudzenia modułu Z-Wave z trybu uśpienia

D. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów komunikacji z modułem w sieci Z-Wave.

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` – komunikacja z modułem nie jest zablokowana, `1` – zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie – jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` *nie jest jednoznaczne z ponowną poprawną komunikacją z modułem – umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

16. Remotec ZXT-310

Wersja modułu: ***ZXT-310EU HW: 00 FW: 1.10***

16.1. Informacje ogólne

Obsługa modułu Remotec ZXT-310 obejmuje możliwość nauki a także wysyłania kodu IR, definiowanie parametrów transmisji oraz odczyt stanu uczenia danego kodu przez urządzenie. Możliwe jest również definiowanie okresu wybudzania modułu.

Sposób dodawania / usuwania: 1x kliknąć przycisk *PROG* w module podczas inclusion / exclusion - czerwona dioda zamruga 1x, a następnie zacznie świecić światłem ciągłym. Jeśli dioda zamruga 6x, to oznacza, że proces dodawania się nie udał.

Sposób przywracania urządzenia do ustawień fabrycznych: należy przytrzymać przycisk *PROG* na urządzeniu przez 10 sekund. Po zakończeniu procedury czerwona dioda powinna zgasnąć i ponownie się zaświecić.

Port 1 to wewnętrzne diody IR urządzenia. Porty 2-6 to zewnętrzne porty IR urządzenia, do których podłącza się dołączone do zestawu przewody z nadajnikami IR.

16.2. Opis konfiguracji urządzenia

A. Sposób uczenia kodów IR

1. Uczenie kodów odbywa się za pomocą głównego obiektu ZWAVE_IR1
2. Wybrać Endpoint, do którego będą przypisywane kody poprzez wywołanie metody `SetEndpointNumber`. Każdy Endpoint posiada reprezentację w postaci obiektu (ZWAVE_IR_EP1, ... ZWAVE_IR_EP6)
3. Wywołać metodę `LearnCode` podając numer kodu IR z przedziału 1-384 pod jakim chcemy, aby kod się zapisał. Po wywołaniu metody, dioda na urządzeniu powinna zgasnąć i zaświecić się ponownie.
4. W ciągu 15 sekund nacisnąć i trzymać przycisk pilota, którego chcemy nauczyć kierując pilot w kierunku oznaczenia „L" na obudowie urządzenia w odległości 1-3 cm.
    - W przypadku poprawnego zaprogramowania kodu IR dioda na urządzeniu powinna mrugnąć 2x.
    - W przypadku niepowodzenia dioda na urządzeniu powinna mrugnąć 6x.

Status uczenia można także odczytać z parametru `LearningStatus`. Dodatkowo generowane są odpowiednie zdarzenia w zależności od statusu uczenia (`OnLearning`, `OnLearningOK`, `OnLearningFail`, `OnCommandFull`)

Uczenie kodów trzeba wykonać dla każdego endpointu osobno. Maksymalna ilość kodów możliwa do zapamiętania to 6*64.

> **Uwaga!** Pozycja pilota względem urządzenia w czasie uczenia jest kluczowa. Zaleca się, aby pilot był nieruchomy względem urządzenia podczas naciskania przycisku. Niepoprawna pozycja może spowodować, że zapamiętany kod będzie błędny mimo poprawnego statusu uczenia.

> **Uwaga!** Pamięć nauczonych kodów jest zachowywana po odłączeniu zasilania urządzenia. Pamięć ta jest czyszczona po zmianie numeru urządzenia AV oraz po usunięciu urządzenia z sieci Z-Wave.

B. Sposób wysyłania kodów IR

1. Wywołać komendę `SendCode` podając numer nauczonego kodu IR z przedziału 1-384.
2. Po wywołaniu metody dioda na urządzeniu powinna zgasnąć i zaświecić się ponownie, a przypisany kod jest wysłany do urządzenia docelowego.

> **Uwaga!** Wysyłanie kodów można wykonywać dla każdego z sześciu endpointów bezpośrednio wybierając jeden z obiektów ZWAVE_IR_EP bądź pośrednio wybierając obiekt ZWAVE_IR i odpowiednio konfigurując numer endpointu w tym obiekcie.

C. Sposób konfiguracji endpointów

Endpointy (ZWAVE_IR_EP1, ZWAVE_IR_EP2, itd.) można konfigurować na dwa sposoby:

- pośrednio poprzez wspólny obiekt ZWAVE_IR - w tym przypadku w pierwszej kolejności należy ustawić numer endpointu, który będzie konfigurowany przy pomocy metody `SetEndpointNumber`.
- bezpośrednio poprzez indywidualne obiekty ZWAVE_IR_EP odpowiadające poszczególnym endpointom. W przypadku wspólnego obiektu ZWAVE_IR

Każdemu z endpointów można przypisać inny port IR. Dostępnych jest 6 portów IR. Domyślnie do wszystkich endpointów jest przypisany port 1. Port 1 to wewnętrzne diody IR urządzenia. Porty 2-6 to zewnętrzne porty IR urządzenia, do których podłącza się dołączone do zestawu przewody z nadajnikami IR.

Po przypisaniu portu IR do danego endpointu, można ustawić pozostałe parametry takie jak moc nadajnika IR (tylko zewnętrzne nadajniki) i tryb transmisji.

> **Uwaga!** Zewnętrzne nadajniki mają bardzo małą moc i mały kąt świecenia, dlatego powinny być umieszczone blisko odbiornika IR kontrolowanego urządzenia i odpowiednio skierowane. Kierunek świecenia nadajników IR jest zgodny z osią przewodu wchodzącego do obudowy nadajnika IR.

> **Uwaga!** Zaleca się nie zmieniać numeru urządzenia AV (cecha `AvDeviceNumber`) jeśli nie korzystamy z wewnętrznej bazy kodów IR urządzenia.

16.3. Obiekty

A. ZWAVE_IR

Obiekt umożliwia odczyt i zapis parametrów konfiguracyjnych wybranego wcześniej endpointu oraz wysyłanie kodów IR przez ten zdefiniowany endpoint.

CECHY

Nazwa	Opis
`PortRouting`	Zwraca numer portu IR przypisanego do aktualnie wybranego endpointu (1 - wewnętrzny port IR, 2 ÷ 6 - zewnętrzne porty IR)
`AvDeviceNumber`	Zwraca numer urządzenia AV z wewnętrznej biblioteki kodów IR przypisanego do aktualnie wybranego endpointu (czterocyfrowy numer z Listy Kodów ZXT-310)
`EmitterPower`	Zwraca moc zewnętrznego nadajnika podczerwieni ustawionego portu IR: `0` - normalna moc `255` - duża moc **Uwaga!** *Parametr* `EmmitterPower` *nie jest konfigurowalny dla portu 1*
`TransmissionMode`	Zwraca tryb transmisji kodu IR: `0` - transmisja ciągła, `255` - pojedynczy impuls
`EndpointNumber`	Zwraca numer kontrolowanego endpointu (1 ÷6)
`FirmwareVersion`	Zwraca numer wersji oprogramowania
`LibraryVersion`	Zwraca numer wersji wbudowanej biblioteki kodów IR
`LearningStatus`	Zwraca status trybu uczenia kodów IR: `0` - kanał IR bezczynny, `1` - uczenie zakończone powodzeniem, `2` - trwa procedura uczenia, `3` - osiągnięto maksymalną ilość kodów dla danego Endpointa, `4` - uczenie zakończone niepowodzeniem

Uwaga! Wartość ustawionych parametrów konfiguracyjnych jest odświeżana w momencie `WakeUp`'u danego urządzenia (pobierane są wartości od urządzenia Z-Wave). Na czas konfigurowania parametrów urządzenia (`SetAvDeviceNumber`, `SetEmitterPower`, `SetTransmissionMode`, `SetPortRouting`) oraz poprawny odczyt ustawionych cech, możliwe jest ustawienie czasu `WakeUpInterval` na czas mniejszy niż 60s. Po dokonaniu zmian i zakończeniu konfiguracji powyższych parametrów należy zmienić czas wybudzania na wartość co najmniej 60s.

METODY

Nazwa	Opis
`SendCode`	Wysyła kod IR o określonym numerze (numer kodu z zakresu 1-384, nauczony lub dostępny w wewnętrznej bibliotece kodów IR dla danego urządzenia AV)
`LearnCode`	Wywołuje tryb uczenia kodu IR o określonym numerze (numer kodu z zakresu 1-384)
`SetPortRouting`	Ustawia numer portu IR, który ma być przypisany do aktualnie wybranego endpointu
`SetAvDeviceNumber`	Ustawia numer urządzenia AV z wewnętrznej biblioteki kodów IR przypisanego do aktualnie wybranego endpointu (czterocyfrowy numer z Listy Kodów ZXT-310)
`SetEmitterPower`	Ustawia moc zewnętrznego nadajnika podczerwieni **Uwaga!** *Parametr* `EmmitterPower` *nie jest konfigurowalny dla portu 1*
`SetTransmissionMode`	Ustawia tryb transmisji kodu IR
`SetEndpointNumber`	Ustawia numer endpointu, który ma być kontrolowany (1 ÷ 6)

ZDARZENIA

Nazwa	Opis
`OnIrSend`	Zdarzenie wywoływane w momencie wysłania kodu IR
`OnLearningStatusChange`	Zdarzenie wywoływane w momencie zmiany statusu trybu uczenia kodu IR
`OnLearningOK`	Zdarzenie wywoływane w momencie zmiany statusu trybu uczenia kodu IR na „*OK*”
`OnLearning`	Zdarzenie wywoływane w momencie zmiany statusu trybu uczenia kodu IR na „*Learning*”
`OnLearning`	Zdarzenie wywoływane w momencie zmiany statusu trybu uczenia kodu IR na „*Command Full*”
`OnLearningFail`	Zdarzenie wywoływane w momencie zmiany statusu trybu uczenia kodu IR na „*Learning Fail*”

B. ZWAVE_IR_EP

Obiekt umożliwia bezpośredni odczyt i zapis parametrów konfiguracyjnych endpointu, do którego się odnosi oraz wysyłanie kodów IR przez ten endpoint. Domyślnie do wszystkich endpointów jest przypisany port 1 (wartość cechy `PortRouting`).

> **Uwaga!** Aby każdy kolejny obiekt (ZWAVE_IR_EP1, ZWAVE_IR_EP2, itd.) odnosił się do kolejnego portu urządzenia (1-6) należy w pierwszej kolejności ustawić cechę `PortRouting`, przykładowo:

ZWAVE_IR_EP1 - `PortRouting`: 1 ZWAVE_IR_EP2 - `PortRouting`: 2 ... ZWAVE_IR_EP6 - `PortRouting`: 6

następnie należy wysłać konfigurację.

CECHY

Nazwa	Opis
`PortRouting`	Zwraca numer portu IR przypisanego do endpointu (1 - wewnętrzny port IR, 2 ÷ 6 - zewnętrzne porty IR)
`AvDeviceNumber`	Zwraca numer urządzenia AV z wewnętrznej biblioteki kodów IR przypisanego do endpointu (czterocyfrowy numer z Listy Kodów ZXT-310)
`EmitterPower`	Zwraca moc zewnętrznego nadajnika podczerwieni ustawionego portu IR: `0` - normalna moc `255` - duża moc **Uwaga!** *Parametr EmmitterPower nie jest konfigurowalny dla portu 1*
`TransmissionMode`	Zwraca tryb transmisji kodu IR: `0` - transmisja ciągła, `255` - pojedynczy impuls

METODY

Nazwa	Opis
`SendCode`	Wysyła kod IR o określonym numerze (numer kodu z zakresu 1-465, nauczony lub dostępny w wewnętrznej bibliotece kodów IR dla danego urządzenia AV)
`SetPortRouting`	Ustawia numer portu IR, który ma być przypisany do aktualnie wybranego endpointu
`SetAvDeviceNumber`	Ustawia numer urządzenia AV z wewnętrznej biblioteki kodów IR przypisanego do aktualnie wybranego endpointu (czterocyfrowy numer z Listy Kodów ZXT-310)
`SetEmitterPower`	Ustawia moc zewnętrznego nadajnika podczerwieni **Uwaga!** *Parametr `EmmitterPower` nie jest konfigurowalny dla portu 1*
`SetTransmissionMode`	Ustawia tryb transmisji kodu IR

ZDARZENIA

Nazwa	Opis
`OnIrSend`	Zdarzenie wywoływane w momencie wysłania kodu IR

C. ZWAVE_WAKEUP

Obiekt umożliwiający ustawienie oraz odczyt czasu odczytu parametrów modułu Z-Wave. Domyślna wartość ustawiania przez CLU wynosi 3600s (60 minut). Minimalna wartość to 10s, maksymalna 16777200s (około 194 dni). Możliwe jest ustawienie wartości w kroku 5s.

> **Uwaga!** Niezalecane jest ustawianie wartości cechy `WakeUp` mniejszej niż 60s podczas normalnej pracy urządzenia. Zmniejszenie wartości może być przydatne w przypadku 'nauczania' kodów przez urządzenie (generowanie zdarzeń zmiany statusu trybu uczenia, a także odczyt cechy `LearningStatus`), jak również przy ustawianiu parametrów konfiguracyjnych.

CECHY

Nazwa	Opis
`Interval`	Okres samoczynnego wybudzania modułu Z-Wave z trybu uśpienia w sekundach
`LastWakeUp`	Czas ostatniego wybudzenia modułu Z-Wave z trybu uśpienia

METODY

Nazwa	Opis
`SetInterval`	Ustawia okres samoczynnego wybudzania modułu Z-Wave z trybu uśpienia

ZDARZENIA

Nazwa	Opis
`OnWakeUp`	Zdarzenie wywoływane po wykryciu wybudzenia modułu Z-Wave z trybu uśpienia

D. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów komunikacji z modułem w sieci Z-Wave.

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` – komunikacja z modułem nie jest zablokowana, `1` – zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie – jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 30s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` *nie jest jednoznaczne z ponowną poprawną komunikacją z modułem – umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

17. Aeotec Nano Switch

Wersja modułu: ZW116-C

17.1. Informacje ogólne

> **Uwaga!** Obsługa modułu dostępna od wersji CLU 5.06.03 *(build 2043)*.

Obsługa Aeotec Nano Switch obejmuje możliwość włączania / wyłączania, na stałe lub na określony czas, wyjścia modułu a także odczyt jego stanu. Sterowanie modułem możliwe jest poprzez metody obiektu ZWAVE_DOUT lub za pomocą zewnętrznych przełączników podłączonych do wejść Nano Switch. Ich konfiguracja (tryb działania) jest możliwa poprzez zmianę odpowiednich parametrów określonych indywidualnie w instrukcji modułu.

Po Inclusion modułu do jednostki CLU Z-Wave, ustawiane są podstawowe parametry konfigurujące pracę przełączników:

Parametr	Domyślna wartość
`120` - wejście S1	`1` - 2-state switch mode (przycisk bistabilny)
`121` - wejście S2	`1` - 2-state switch mode (przycisk bistabilny)

Działanie wejść S1, S2: wejścia S1, S2 sterują bezpośrednio wyjściem modułu:

- Stan wysoki wejścia -> wyjście załączone,
- Stan niski wejścia -> wyjście wyłączone.

Sposób dodawania / usuwania: W celu dodania / usunięcia modułu należy krótko nacisnąć na przycisk akcji w module Nano Switch podczas Inclusion / Exclusion wywołanych na CLU.

> **Uwaga!** Po restarcie (wysłaniu konfiguracji) CLU, należy odczekać 10 sekund przed pierwszą próbą załączenia modułu Nano Switch.

17.2. Obiekty

A. ZWAVE_DOUT

Obiekt umożliwiający włączanie / wyłączanie urządzenia na stałe lub na określony czas oraz odczyt jego aktualnego stanu.

CECHY

Nazwa	Opis
`Value`	Zwraca 1 dla wyjścia ustawionego na "On" i 0 dla wyjścia ustawionego na "Off"

METODY

Nazwa	Opis
`SetValue`	Ustawia stan wyjścia jako 0 lub 1
`Switch`	Zmienia stan wyjścia na przeciwny. Parametr Time określa na jak długo następuje zmiana stanu, dla 0 jest ona stała
`SwitchOn`	Załącza wyjście. Parametr Time określa na jak długo następuje zmiana stanu, dla 0 jest ona stała
`SwitchOff`	Wyłącza wyjście. Parametr Time określa na jak długo następuje zmiana stanu, dla 0 jest ona stała

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w przypadku zmiany stanu na przeciwny
`OnSwitchOn`	Zdarzenie wywoływane w momencie ustawienia stanu wysokiego na wyjściu
`OnSwitchOff`	Zdarzenie wywoływane w momencie ustawienia stanu niskiego na wyjściu

B. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów komunikacji z modułem w sieci Z-Wave. Umożliwia ustawienie zaawansowanych parametrów konfiguracyjnych danego modułu.

> **Uwaga!** Zmiana parametrów konfiguracyjnych jest możliwa dopiero po ustawieniu parametru 252 na wartość 0 (Unlock), domyślnie ustawiony na 1 (Lock).

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` – komunikacja z modułem nie jest zablokowana, `1` – zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie – jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)
`Register`	Numer rejestru (parametru) konfiguracyjnego, który ostatnio został odczytany/ustawiony za pomocą dostępnych metod
`Value`	Wartość rejestru (parametru) konfiguracyjnego

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** *`RemoveBan` nie jest jednoznaczne z ponowną poprawną komunikacją z modułem – umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji
`Set`	Ustawia wartość danego rejestru (parametru) konfiguracyjnego: `Register` (numer rejestru lub parametru), `Value` (wartość rejestru bądź parametru), `Size` (rozmiar wysyłanej wartości rejestru lub parametru – w bajtach)
`Get`	Pobiera wartość danego rejestru (parametru) konfiguracyjnego
`SetDefault`	Ustawia wartość domyślną dla danego rejestru (parametru) konfiguracyjnego

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

18. Aeotec Dual Nano Switch

Wersja modułu: ZW140-C

18.1. Informacje ogólne

> **Uwaga!** Obsługa modułu dostępna od wersji CLU 5.06.03 *(build 2043)*.

Obsługa Aeotec Dual Nano Switch obejmuje możliwość włączania / wyłączania, na stałe lub na określony czas, wyjść modułu a także odczyt ich stanu. Sterowanie modułem możliwe jest poprzez metody obiektów ZWAVE_DOUT lub za pomocą zewnętrznych przełączników podłączonych do wejść Dual Nano Switch. Ich konfiguracja (tryb działania) jest możliwa poprzez zmianę odpowiednich parametrów określonych indywidualnie w instrukcji modułu.

Po Inclusion modułu do jednostki CLU Z-Wave, ustawiane są podstawowe parametry konfigurujące pracę przełączników:

Parametr	Domyślna wartość
`120` - wejście S1	`1` - 2-state switch mode (przycisk bistabilny)
`121` - wejście S2	`1` - 2-state switch mode (przycisk bistabilny)

Działanie wejść S1, S2: wejścia S1, S2 sterują bezpośrednio wyjściami modułu (OUT1, OUT2), odpowiednio S1 -> OUT1, S2 -> OUT2:

- Stan wysoki wejścia -> wyjście załączone,
- Stan niski wejścia -> wyjście wyłączone.

Sposób dodawania / usuwania: W celu dodania / usunięcia modułu należy krótko nacisnąć na przycisk akcji w module Dual Nano Switch podczas Inclusion / Exclusion wywołanych na CLU.

> **Uwaga!** Po restarcie (wysłaniu konfiguracji) CLU, należy odczekać 10 sekund przed pierwszą próbą załączenia modułu Dual Nano Switch.

18.2. Obiekty

A. ZWAVE_DOUT

Obiekt umożliwiający włączanie / wyłączanie urządzenia na stałe lub na określony czas oraz odczyt jego aktualnego stanu.

CECHY

Nazwa	Opis
`Value`	Zwraca 1 dla wyjścia ustawionego na "On" i 0 dla wyjścia ustawionego na "Off"

METODY

Nazwa	Opis
`SetValue`	Ustawia stan wyjścia jako 0 lub 1
`Switch`	Zmienia stan wyjścia na przeciwny. Parametr Time określa na jak długo następuje zmiana stanu, dla 0 jest ona stała
`SwitchOn`	Załącza wyjście. Parametr Time określa na jak długo następuje zmiana stanu, dla 0 jest ona stała
`SwitchOff`	Wyłącza wyjście. Parametr Time określa na jak długo następuje zmiana stanu, dla 0 jest ona stała

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w przypadku zmiany stanu na przeciwny
`OnSwitchOn`	Zdarzenie wywoływane w momencie ustawienia stanu wysokiego na wyjściu
`OnSwitchOff`	Zdarzenie wywoływane w momencie ustawienia stanu niskiego na wyjściu

B. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów komunikacji z modułem w sieci Z-Wave. Umożliwia ustawienie zaawansowanych parametrów konfiguracyjnych danego modułu.

> **Uwaga!** Zmiana parametrów konfiguracyjnych jest możliwa dopiero po ustawieniu parametru 252 na wartość 0 (Unlock), domyślnie ustawiony na 1 (Lock).

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` – komunikacja z modułem nie jest zablokowana, `1` – zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie – jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)
`Register`	Numer rejestru (parametru) konfiguracyjnego, który ostatnio został odczytany/ustawiony za pomocą dostępnych metod
`Value`	Wartość rejestru (parametru) konfiguracyjnego

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** *`RemoveBan` nie jest jednoznaczne z ponowną poprawną komunikacją z modułem – umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji
`Set`	Ustawia wartość danego rejestru (parametru) konfiguracyjnego: `Register` (numer rejestru lub parametru), `Value` (wartość rejestru bądź parametru), `Size` (rozmiar wysyłanej wartości rejestru lub parametru – w bajtach)
`Get`	Pobiera wartość danego rejestru (parametru) konfiguracyjnego
`SetDefault`	Ustawia wartość domyślną dla danego rejestru (parametru) konfiguracyjnego

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

19. Aeotec Nano Dimmer

Wersja modułu: ZW111-C

19.1. Informacje ogólne

> **Uwaga!** Obsługa modułu dostępna od wersji CLU 5.06.03 *(build 2043)*.

Obsługa Aeotec Nano Dimmer obejmuje możliwość płynnego sterowania poziomem natężenia światła oraz odczyt tej wartości. Sterowanie oświetleniem możliwe jest poprzez metody obiektu ZWAVE_DIMMER lub za pomocą zewnętrznych przełączników podłączonych do wejść Nano Dimmer. Ich konfiguracja (tryb działania) jest możliwa poprzez zmianę odpowiednich parametrów określonych indywidualnie w instrukcji modułu.

Po Inclusion modułu do jednostki CLU Z-Wave, ustawiane są podstawowe parametry konfigurujące pracę przełączników:

Parametr	Domyślna wartość
`120` - wejście S1	`3` - momentary push button mode (przycisk monostabilny)
`121` - wejście S2	`1` - 2-state switch mode (przycisk bistabilny)
`125` - szybkość ściemniania w sekundach	`3`

Działanie wejścia S1: wejście S1 steruje bezpośrednio wyjściem modułu:

- Pojedyncze kliknięcie -> ustawia ostatnią zapamiętaną wartość ściemniacza / wyłącza ściemniacz,
- Naciśnięcie i przytrzymanie -> zwiększenie lub zmniejszenie wartości ściemniacza.

Działanie wejścia S2: wejście S2 powiązane jest z obiektem ZWAVE_DIN, nie steruje bezpośrednio wyjściem modułu. Dokładne działanie znajduje się w opisie obiektu ZWAVE_DIN.

Sposób dodawania / usuwania: W celu dodania / usunięcia modułu należy krótko nacisnąć na przycisk akcji w module Nano Dimmer podczas Inclusion / Exclusion wywołanych na CLU.

> **Uwaga!** Po restarcie (wysłaniu konfiguracji) CLU, należy odczekać 10 sekund przed pierwszą próbą załączenia modułu Nano Dimmer.

19.2. Obiekty

A. ZWAVE_DIMMER

Obiekt umożliwiający sterowanie poziomem natężenia światła oraz odczyt aktualnego stanu urządzenia.

CECHY

Nazwa	Opis
`Value`	Aktualna wartość ściemniacza
`MinValue`	Minimalna wartość ściemniacza w procentach
`MaxValue`	Maksymalna wartość ściemniacza w procentach
`RampTime`	Czas narastania/opadania zmiany wartości ściemniacza w milisekundach

METODY

Nazwa	Opis
`SetValue`	Ustawia wartość ściemniacza
`SetMinValue`	Ustawia minimalną wartość ściemniacza
`SetMaxValue`	Ustawia maksymalną wartość ściemniacza
`SetRampTime`	Ustawia czas narastania/opadania zmiany wartości ściemniacza
`Switch`	Przełącza stan ściemniacza na przeciwny (używając wartości MinValue i MaxValue), Time - czas na jaki ściemniacz ma zostać przełączony (wartość 0 oznacza czas nieograniczony), RampTime - czas narastania/opadania zmiany wartości (wartość 0 oznacza użycie cechy RampTime)
`SwitchOn`	Przełącza stan ściemniacza na włączony (używając wartości MaxValue), Time - czas na jaki ściemniacz ma zostać przełączony (wartość 0 oznacza czas nieograniczony), RampTime - czas narastania/opadania zmiany wartości (wartość 0 oznacza użycie cechy RampTime)
`SwitchOff`	Przełącza stan ściemniacza na wyłączony (używając wartości MinValue), Time - czas na jaki ściemniacz ma zostać przełączony (wartość 0 oznacza czas nieograniczony), RampTime - czas narastania/opadania zmiany wartości (wartość 0 oznacza użycie cechy RampTime)
`Hold`	Rozpoczyna i przetrzymuje akcje płynnego rozjaśniania/ściemniania w kierunku przeciwnym do poprzedniego, RampTime - czas rozjaśniania/ściemniania do wartości brzegowej (domyślnie 15000ms)
`HoldUp`	Rozpoczyna i przetrzymuje akcje płynnego rozjaśniania, RampTime - czas rozjaśniania do wartości brzegowej (domyślnie 15000ms)
`HoldDown`	Rozpoczyna i przetrzymuje akcje płynnego ściemniania, RampTime - czas ściemniania do wartości brzegowej (domyślnie 15000ms)

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane przy zmianie wartości ściemniacza
`OnSwitchOn`	Zdarzenie wywoływane przy zmianie stanu ściemniacza na włączony (wartość większa niż MinValue)
`OnSwitchOff`	Zdarzenie wywoływane przy zmianie stanu ściemniacza na wyłączony (wartość mniejsza lub równa MinValue)
`OnValueRise`	Zdarzenie wywoływane przy wzroście wartości ściemniacza
`OnValueDrop`	Zdarzenie wywoływane przy spadku wartości ściemniacza

B. ZWAVE_DIN

Działanie obiektu jest determinowane ustawieniem parametru konfiguracyjnego `121` modułu Nano Dimmer, który określa tryb pracy dla wejścia S2:

- dla trybu wejścia **2 - State switch mode**:
  - Stan wysoki wejścia -> cecha wbudowana `Value` przyjmuje wartość 1,
  - Stan niski wejścia -> cecha wbudowana `Value` przyjmuje wartość 0.
- dla trybu wejścia **Momentary push button mode**:
  - Pojedyncze kliknięcie -> cecha wbudowana `Value` przyjmuje wartość 1,
  - Ponowne pojedyncze kliknięcie -> cecha wbudowana `Value` przyjmuje wartość 0.

> **Uwaga!** Na podstawie zaprezentowanych powyżej opisów zmiany cechy wbudowanej `Value` obiektu ZWAVE_DIN należy odpowiednio dostosować konfigurację w powiązaniu ZWAVE_DIN -> ZWAVE_DIMMER w celu osiągnięcia pożądanej funkcjonalności (załącz / wyłącz, ściemnianie). W przypadku chęci wykonania standardowej konfiguracji w systemie Grenton do sterowania obiektem DIMMER należy wykorzystać adekwatny moduł wejściowy DIN lub TouchPanel.

CECHY

Nazwa	Opis
`HoldDelay`	Czas w milisekundach, po jakim po wciśnięciu i przytrzymaniu przycisku wyzwalane jest zdarzenie `OnHold`
`HoldInterval`	Odstęp cykliczny w milisekundach, po jakim podczas trzymania przycisku wyzwalane jest zdarzenie `OnHold`
`Value`	Zwraca stan przycisku jako 0 lub 1

METODY

Nazwa	Opis
`SetHoldDelay`	Ustawia wartość `HoldDelay`
`SetHoldInterval`	Ustawia wartość `HoldInterval`

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane przy zmianie stanu na przeciwny
`OnSwitchOn`	Zdarzenie wywoływane w momencie ustawienia stanu wysokiego na wejściu
`OnSwitchOff`	Zdarzenie wywoływane w momencie ustawienia stanu niskiego na wejściu
`OnShortPress`	Zdarzenie wywoływane po naciśnięciu przycisku na czas 500 ms - 2000 ms
`OnLongPress`	Zdarzenie wywoływane po naciśnięciu przycisku na czas dłuższy niż 2000 ms
`OnHold`	Zdarzenie wywoływane gdy wejście jest w stanie wysokim, pierwszy raz po upłynięciu czasu `HoldDelay` , a następnie cyklicznie co wartość `HoldInterval`
`OnClick`	Zdarzenie wywoływane po naciśnięciu przycisku na czas krótszy niż 500ms

C. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów komunikacji z modułem w sieci Z-Wave. Umożliwia ustawienie zaawansowanych parametrów konfiguracyjnych danego modułu.

> **Uwaga!** Zmiana parametrów konfiguracyjnych jest możliwa dopiero po ustawieniu parametru 252 na wartość 0 (Unlock), domyślnie ustawiony na 1 (Lock).

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` - komunikacja z modułem nie jest zablokowana, `1` - zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie - jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)
`Register`	Numer rejestru (parametru) konfiguracyjnego, który ostatnio został odczytany/ustawiony za pomocą dostępnych metod
`Value`	Wartość rejestru (parametru) konfiguracyjnego

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` *nie jest jednoznaczne z ponowną poprawną komunikacją z modułem - umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji
`Set`	Ustawia wartość danego rejestru (parametru) konfiguracyjnego: `Register` (numer rejestru lub parametru), `Value` (wartość rejestru bądź parametru), `Size` (rozmiar wysyłanej wartości rejestru lub parametru - w bajtach)
`Get`	Pobiera wartość danego rejestru (parametru) konfiguracyjnego
`SetDefault`	Ustawia wartość domyślną dla danego rejestru (parametru) konfiguracyjnego

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

20. Aeotec Nano Shutter

Wersja modułu: ZW141-C

20.1. Informacje ogólne

> **Uwaga!** Obsługa modułu dostępna od wersji CLU 5.06.03 *(build 2043)*.

Obsługa Aeotec Nano Shutter obejmuje możliwość sterowania wyjściami modułu (góra / dół / stop). Sterowanie modułem możliwe jest poprzez metody obiektu ZWAVE_ROLLER_SHUTTER lub za pomocą zewnętrznych przełączników podłączonych do wejść Nano Shutter. Ich konfiguracja (tryb działania) jest możliwa poprzez zmianę odpowiednich parametrów określonych indywidualnie w instrukcji modułu.

Po Inclusion modułu do jednostki CLU Z-Wave, ustawiane są podstawowe parametry konfigurujące pracę zewnętrznych przełączników:

Parametr	Domyślna wartość
`35` - czas przejazdu rolety w sekundach	`11`
`120` - wejście S1	`1` - 2-state switch mode (przycisk bistabilny)
`121` - wejście S2	`3` - momentary push button mode (przycisk monostabilny)

Działanie wejścia S1: wejście S1 steruje bezpośrednio wyjściami modułu:

- Stan wysoki na wejściu -> roleta do góry (wyjście OUT1 załączone) jeśli poprzednio ruch w dół / roleta w dół (wyjście OUT2 załączone) jeśli poprzednio ruch w górę,
- Stan niski na wejściu -> zatrzymanie rolety, wyjścia wyłączone .

Działanie wejścia S2: wejście S2 steruje bezpośrednio wyjściami modułu:

- Pojedyncze kliknięcie -> roleta do góry (wyjście OUT1 załączone) jeśli poprzedni ruch w dół / zatrzymanie rolety, wyjścia wyłączone / roleta w dół (wyjście OUT2 załączone) jeśli poprzednio ruch w górę.

Sposób dodawania / usuwania: W celu dodania modułu należy krótko nacisnąć na przycisk akcji w module Nano Shutter podczas Inclusion wywołanego na CLU, aby usunąć moduł należy krótko nacisnąć na przycisk akcji 6 razy podczas Exclusion.

> **Uwaga!** Po restarcie (wysłaniu konfiguracji) CLU, należy odczekać 10 sekund przed pierwszą próbą załączenia modułu Nano Shutter.

20.2. Obiekty

A. ZWAVE_ROLLER_SHUTTER

Obiekt umożliwiający sterowanie roletą (góra / dół / stop). Stan rolety określany jest na podstawie wywoływanych metod.

> **Uwaga!** Obiekt nie przejmuje informacji o rzeczywistym stanie urządzenia sterowanego za pomocą zewnętrznych przełączników podłączonych do wejść S1, S2.

> **Uwaga!** Dla prawidłowego działania obiektu należy cechę `MaxTime` oraz parametr konfiguracyjny `35` modułu Nano Shutter ustawić na taką samą wartość, uwzględniając poprawną zamianę jednostek czasu.

CECHY

Nazwa	Opis
`OUT1`	Stan przekaźnika OUT1
`OUT2`	Stan przekaźnika OUT2
`State`	Stan wyjścia: `0` - stoi, `1` - ruch w górę, `2` - ruch w dół
`MaxTime`	Domyślna wartość parametru Time, jeśli wpisano 0

METODY

Nazwa	Opis
`Up`	Roleta do góry. Parametr Time to czas przez jaki roleta ma się otwierać. `number` - czas otwierania, `0` - czas otwierania równy MaxTime
`Down`	Roleta w dół. Parametr Time to czas przez jaki roleta ma się zamykać. `number` - czas zamykania, `0` - czas zamykania równy MaxTime
`Start`	Roleta do góry jeśli poprzednio ruch w dół, roleta w dół jeśli poprzednio ruch w góre. Parametr Time to czas przez jaki roleta ma byc w ruchu. `number` - czas jazdy rolety, `0` - czas jazdy równy MaxTime
`Stop`	Stop jeśli roleta jest w ruchu
`Hold`	Hold z odwracaniem kierunku
`HoldUp`	Hold zawsze w górę
`HoldDown`	Hold zawsze w dół

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w momencie zmiany stanu wyjścia
`OnUp`	Zdarzenie wywoływane w momencie zmiany stanu ze Stop na Up
`OnDown`	Zdarzenie wywoływane w momencie zmiany stanu ze Stop na Down
`OnStart`	Zdarzenie wywoływane w momencie wywołania komendy Start
`OnStop`	Zdarzenie wywoływane w momencie wywołania komendy Stop

B. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów komunikacji z modułem w sieci Z-Wave. Umożliwia ustawienie zaawansowanych parametrów konfiguracyjnych danego modułu.

> **Uwaga!** Zmiana parametrów konfiguracyjnych jest możliwa dopiero po ustawieniu parametru 252 na wartość 0 (Unlock), domyślnie ustawiony na 1 (Lock).

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` - komunikacja z modułem nie jest zablokowana, `1` - zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie - jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)
`Register`	Numer rejestru (parametru) konfiguracyjnego, który ostatnio został odczytany/ustawiony za pomocą dostępnych metod
`Value`	Wartość rejestru (parametru) konfiguracyjnego

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` *nie jest jednoznaczne z ponowną poprawną komunikacją z modułem - umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji
`Set`	Ustawia wartość danego rejestru (parametru) konfiguracyjnego: `Register` (numer rejestru lub parametru), `Value` (wartość rejestru bądź parametru), `Size` (rozmiar wysyłanej wartości rejestru lub parametru - w bajtach)
`Get`	Pobiera wartość danego rejestru (parametru) konfiguracyjnego
`SetDefault`	Ustawia wartość domyślną dla danego rejestru (parametru) konfiguracyjnego

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

21. Aeotec Nano Shutter (V2)

Wersja modułu: ZW141-C

21.1. Informacje ogólne

> **Uwaga!** Obsługa modułu dostępna od wersji CLU 5.08.01 *(build 2128B)*.

Obsługa Aeotec Nano Shutter obejmuje możliwość sterowania wyjściami modułu (góra / dół / stop). Sterowanie modułem możliwe jest poprzez metody obiektu ZWAVE_ROLLER_SHUTTER lub za pomocą zewnętrznych przełączników podłączonych do wejść Nano Shutter. Ich konfiguracja (tryb działania) jest możliwa poprzez zmianę odpowiednich parametrów określonych indywidualnie w instrukcji modułu.

> **Uwaga!** Mogą występować liczne opóźnienia i anomalie wykonywania poleceń przez moduły, zwłaszcza będące w zasięgu działania innych urządzeń typu Z-WAVE lub urządzeń korzystających z częstotliwości radiowej 868.42 MHz jak np.piloty do bram.

Po Inclusion modułu do jednostki CLU Z-Wave, ustawiane są podstawowe parametry konfigurujące pracę zewnętrznych przełączników:

Parametr	Domyślna wartość
`35` - czas przejazdu rolety w setnych częściach sekundy	`15000`
`120` - wejście S1	`1` - 2-state switch mode (przycisk bistabilny)
`121` - wejście S2	`3` - momentary push button mode (przycisk monostabilny)

Działanie wejścia S1: wejście S1 steruje bezpośrednio wyjściami modułu:

- Stan wysoki na wejściu -> roleta do góry (wyjście OUT1 załączone) jeśli poprzednio ruch w dół / roleta w dół (wyjście OUT2 załączone) jeśli poprzednio ruch w górę,
- Stan niski na wejściu -> zatrzymanie rolety, wyjścia wyłączone .

Działanie wejścia S2: wejście S2 steruje bezpośrednio wyjściami modułu:

- Pojedyncze kliknięcie -> roleta do góry (wyjście OUT1 załączone) jeśli poprzedni ruch w dół / zatrzymanie rolety, wyjścia wyłączone / roleta w dół (wyjście OUT2 załączone) jeśli poprzednio ruch w górę.

Sposób dodawania / usuwania: W celu dodania modułu należy krótko nacisnąć na przycisk akcji w module Nano Shutter podczas Inclusion wywołanego na CLU, aby usunąć moduł należy krótko nacisnąć na przycisk akcji 6 razy podczas Exclusion.

> **Uwaga!** Po restarcie (wysłaniu konfiguracji) CLU, należy odczekać 10 sekund przed pierwszą próbą załączenia modułu Nano Shutter.

21.2. Obiekty

A. ZWAVE_ROLLER_SHUTTER

Obiekt umożliwiający sterowanie roletą (góra / dół / stop). Stan rolety określany jest na podstawie wywoływanych metod.

> **Uwaga!** Obiekt nie przejmuje informacji o rzeczywistym stanie urządzenia sterowanego za pomocą zewnętrznych przełączników podłączonych do wejść S1, S2 oraz przycisku znajdującego się na obudowie modułu.

CECHY

Nazwa	Opis
`OUT1`	Stan przekaźnika OUT1
`OUT2`	Stan przekaźnika OUT2
`State`	Stan wyjścia: `0` - stoi, `1` - ruch w górę, `2` - ruch w dół
`Position`	Procentowe określenie otwarcia rolety: `0%` - pełne zamknięcie, `100%` - pełne otwarcie
`MoveTime`	Czas w milisekundach potrzebny do pełnego otwarcia / zamknięcia rolety

METODY

Nazwa	Opis
`Up`	Roleta do góry
`Down`	Roleta w dół
`Start`	Roleta do góry jeśli poprzednio ruch w dół, roleta w dół jeśli poprzednio ruch w góre
`Stop`	Stop jeśli roleta jest w ruchu
`Hold`	Hold z odwracaniem kierunku
`HoldUp`	Hold zawsze w górę
`HoldDown`	Hold zawsze w dół

ZDARZENIA

Nazwa	Opis
`OnChange`	Zdarzenie wywoływane w momencie zmiany stanu wyjścia
`OnUp`	Zdarzenie wywoływane w momencie zmiany stanu ze Stop na Up
`OnDown`	Zdarzenie wywoływane w momencie zmiany stanu ze Stop na Down
`OnStart`	Zdarzenie wywoływane w momencie wywołania komendy Start
`OnStop`	Zdarzenie wywoływane w momencie wywołania komendy Stop

B. ZWAVE_CONFIG

Obiekt wyświetla informacje odnośnie parametrów komunikacji z modułem w sieci Z-Wave. Umożliwia ustawienie zaawansowanych parametrów konfiguracyjnych danego modułu.

> **Uwaga!** Zmiana parametrów konfiguracyjnych jest możliwa dopiero po ustawieniu parametru 252 na wartość 0 (Unlock), domyślnie ustawiony na 1 (Lock).

CECHY

Nazwa	Opis
`NodeID`	Numer modułu (węzła) w sieci Z-Wave (nadawany dla każdego modułu Z-Wave po dodaniu go do kontrolera)
`Banned`	Informacja o zablokowaniu komunikacji Z-Wave z modułem: `0` - komunikacja z modułem nie jest zablokowana, `1` - zablokowana komunikacja z modułem (moduł zbanowany). Zablokowanie następuje w momencie, gdy 3 kolejne próby komunikacji z modułem zakończą się niepowodzeniem (inkrementacja cechy `FailCount` o 3). Do zbanowanego modułu, co 1,5 minuty wysyłane jest zapytanie - jeśli CLU dostanie odpowiedź, wówczas zablokowanie zostanie usunięte i możliwa jest ponowna próba wysłania rozkazu do modułu
`FailCount`	Liczba nieudanych prób komunikacji z modułem Z-Wave. W przypadku niepowodzenia komunikacji z modułem (brak odpowiedzi, potwierdzenia itp.) następuje inkrementacja cechy o 1, następnie próba ponawiania jest dwukrotnie (w interwałach 15s). W przypadku niepowodzenia komunikacja z modułem zostaje zablokowana (`Banned` =1)
`Register`	Numer rejestru (parametru) konfiguracyjnego, który ostatnio został odczytany/ustawiony za pomocą dostępnych metod
`Value`	Wartość rejestru (parametru) konfiguracyjnego

METODY

Nazwa	Opis
`RemoveBan`	Zdejmuje blokadę komunikacji z modułem Z-Wave (w przypadku, gdy cecha `Banned` =1). Wywołanie metody umożliwia ponowne wysłanie rozkazu do modułu. **Uwaga!** `RemoveBan` *nie jest jednoznaczne z ponowną poprawną komunikacją z modułem - umożliwia ponowne wysłanie rozkazu/zapytania do modułu! W przypadku niepowodzenia cały proces blokowania jest rozpoczynany na nowo!*
`ClearFailCount`	Czyści liczbę nieudanych prób komunikacji
`Set`	Ustawia wartość danego rejestru (parametru) konfiguracyjnego: `Register` (numer rejestru lub parametru), `Value` (wartość rejestru bądź parametru), `Size` (rozmiar wysyłanej wartości rejestru lub parametru - w bajtach)
`Get`	Pobiera wartość danego rejestru (parametru) konfiguracyjnego
`SetDefault`	Ustawia wartość domyślną dla danego rejestru (parametru) konfiguracyjnego

ZDARZENIA

Nazwa	Opis
`OnBanned`	Zdarzenie wywoływane, gdy urządzenie zostanie zbanowane

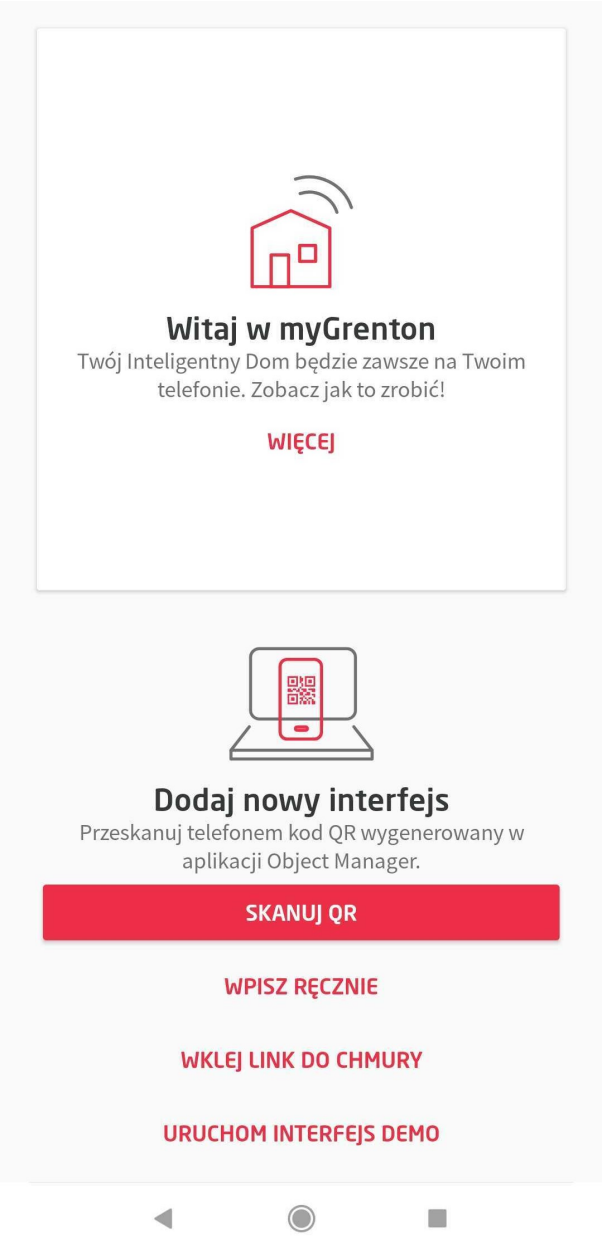
XVIII. Aplikacja mobilna myGrenton

1. Instalacja oraz pierwsze uruchomienie aplikacji myGrenton

1.1. Instalacja

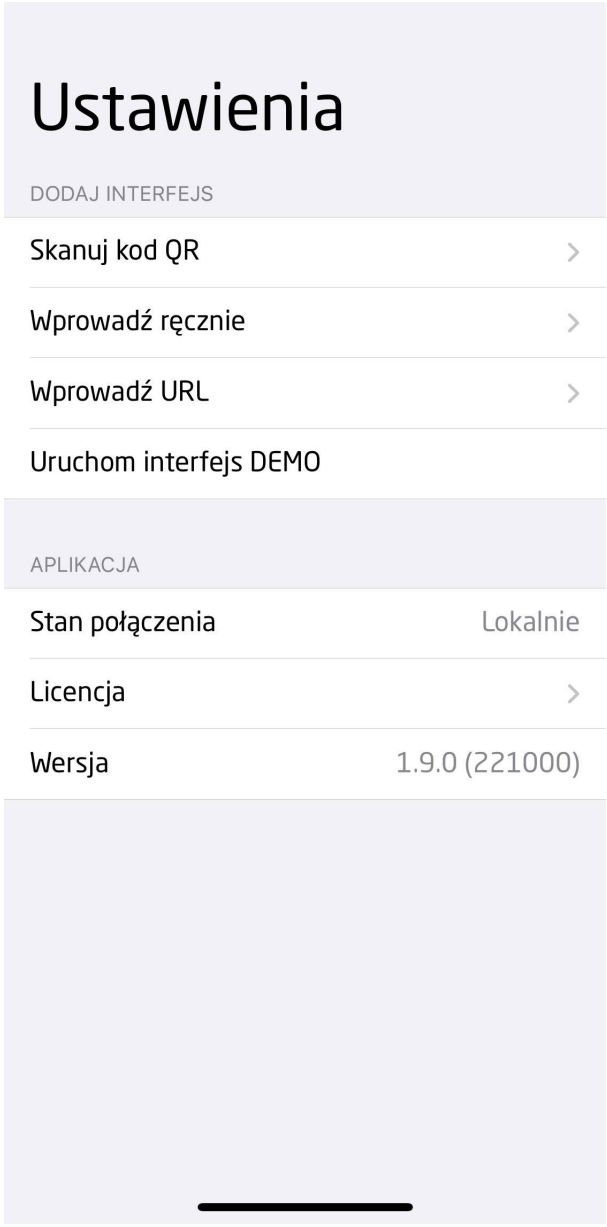
A. Android

Aktualna wersja aplikacji myGrenton dostępna jest w Sklepie Play dla telefonów lub tabletów z systemem **Android (wersja 5.0 lub wyższa)**. Po otwarciu aplikacji pokazuje się okno powitalne z informacjami o aplikacji oraz możliwość dodania nowego interfejsu.



B. iOS

Aktualna wersja aplikacji myGrenton dostępna jest w App Store dla telefonów lub tabletów z systemem **iOS (12 lub wyższy)**. Po otwarciu aplikacji pokazuje się okno z ustawieniami, skąd można m.in. dodać interfejs myGrenton.



Uwaga! Do poprawnego działania aplikacji wymagana minimalna wersja CLU **05.06.04**. Dla wybranych widgetów minimalne wersje CLU oraz Object Manager potrzebne do prawidłowej obsługi zostały podane w ich opisie.

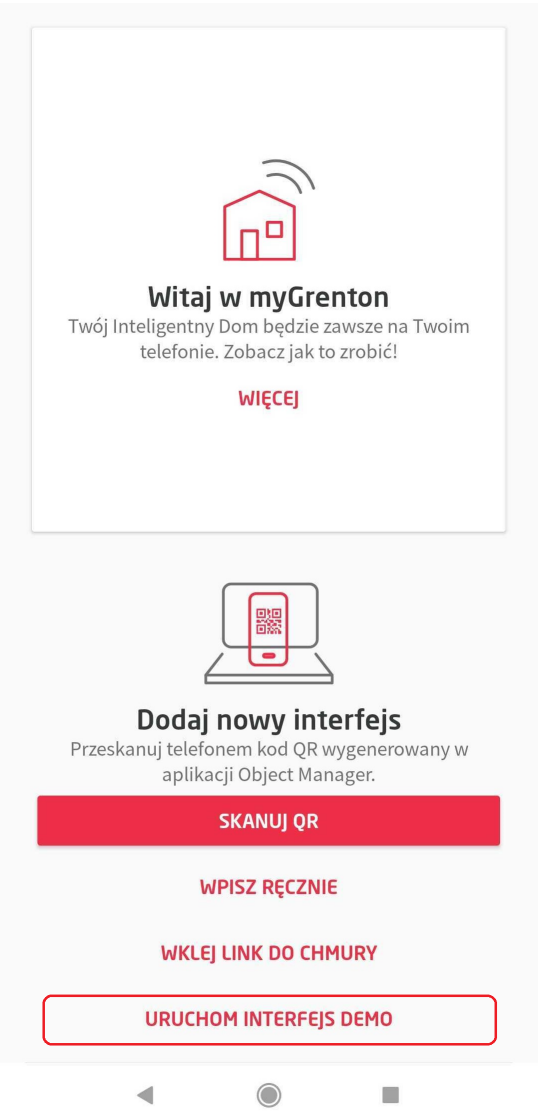
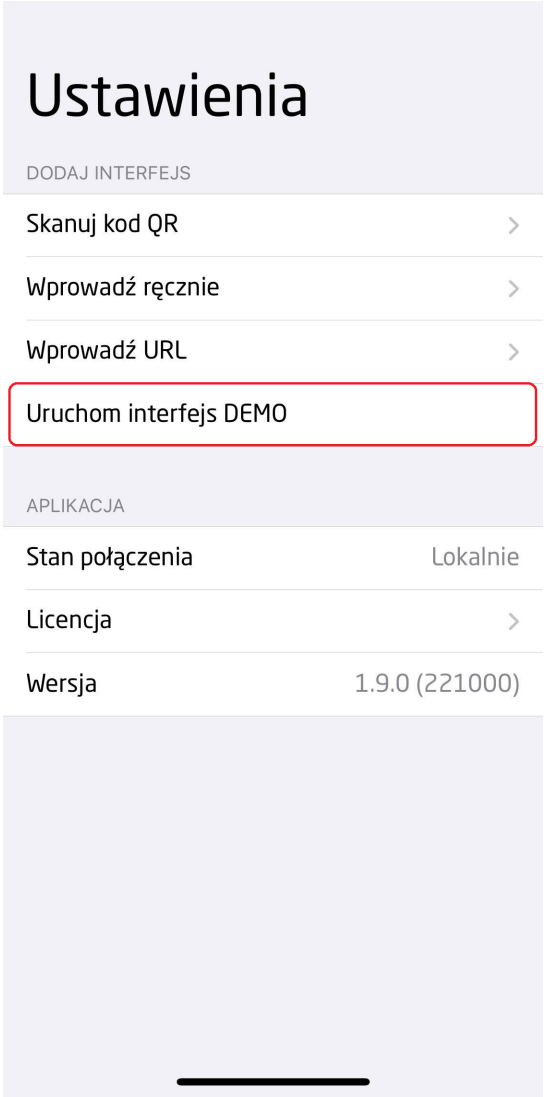
Uwaga! Aplikacja może prosić m.in. o pozwolenie na wykonywanie zdjęć i filmów wideo, pełny dostęp do sieci, wyświetlanie połączeń sieciowych, zapobieganie przejściu telefonu w stan uśpienia, odbieranie danych z internetu. Do poprawnego działania aplikacji należy wyrazić zgodę na powyższe warunki.

1.2. Pierwsze uruchomienie, interfejs demonstracyjny

Aplikacja myGrenton podczas pierwszego uruchomienia pozwala na zapoznanie się z interfejsem aplikacji, możliwościami widgetów i kluczowymi funkcjonalnościami bez połączenia z rzeczywistym systemem Grenton za pomocą interfejsu DEMO.

Dodanie interfejsu DEMO jest możliwe z poziomu widoku pierwszego uruchomienia aplikacji lub kolejnych (widok Ustawienia) tylko w sytuacji, jeśli użytkownik nie dodał interfejsu połączonego z rzeczywistym systemem. Interfejs zawiera kilka przykładowych stron z w pełni funkcjonalnymi widgetami.

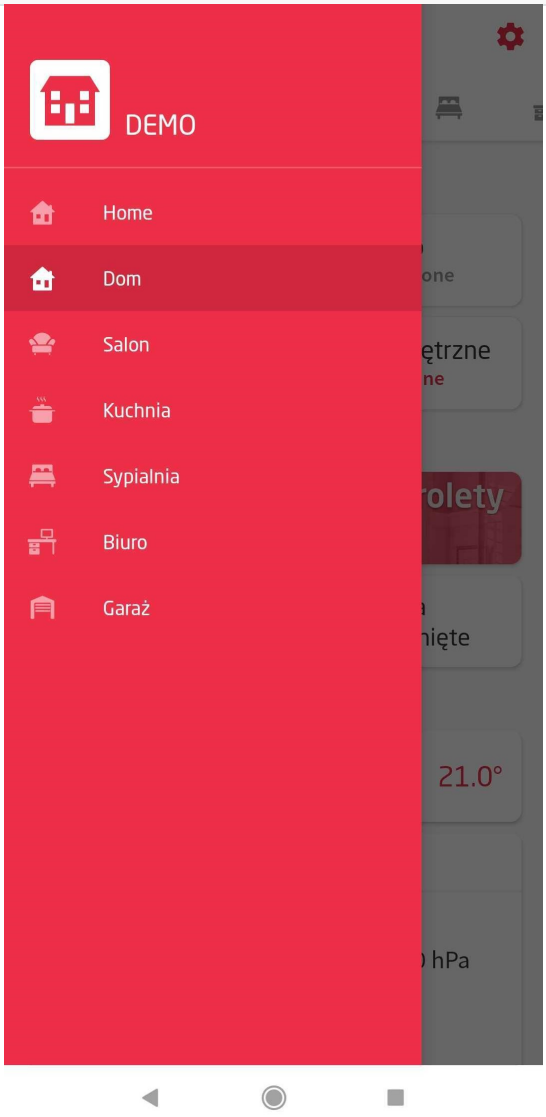
> **Uwaga!** Interfejs DEMO dostępny jest dla aplikacji myGrenton w wersji 1.5.0 (Android) / 1.9.0 (iOS) lub wyższej.

Android	iOS
	

Witaj w myGrenton
Twój Inteligentny Dom będzie zawsze na Twoim telefonie. Zobacz jak to zrobić!
WIĘCEJ
Dodaj nowy interfejs
Przeskanuj telefonem kod QR wygenerowany w aplikacji Object Manager.
SKANUJ QR
WPISZ RĘCZNIE
WKLEJ LINK DO CHMURY
URUCHOM INTERFEJS DEMO

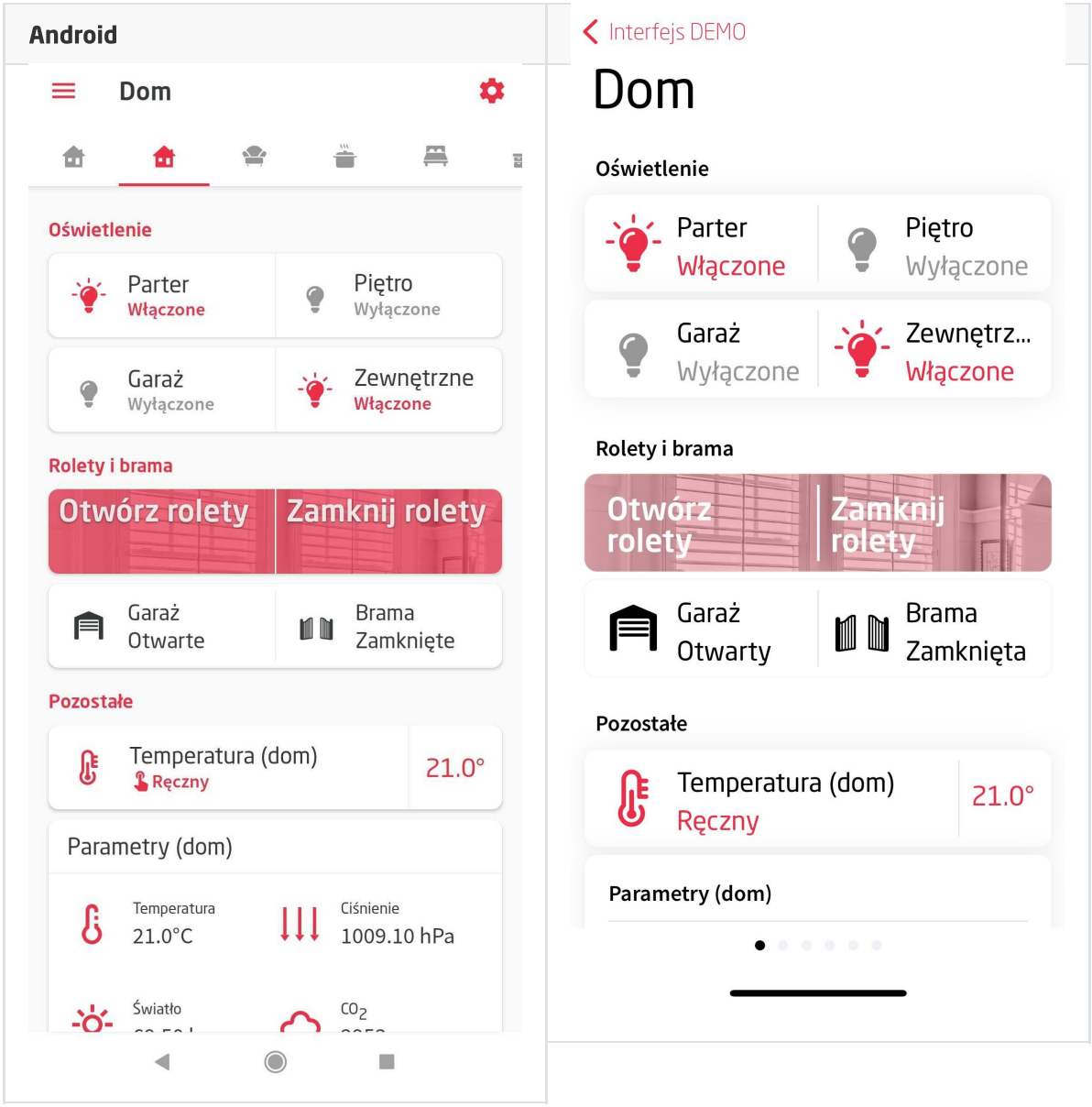
Ustawienia
DODAJ INTERFEJS
Skanuj kod QR
Wprowadź ręcznie
Wprowadź URL
Uruchom interfejs DEMO
APLIKACJA
Stan połączenia
Lokalnie
Licencja
Wersja
1.9.0 (221000)

Android



Interfejs DEMO

- Dom
- Salon
- Kuchnia
- Sypialnia
- Biuro
- Garaż



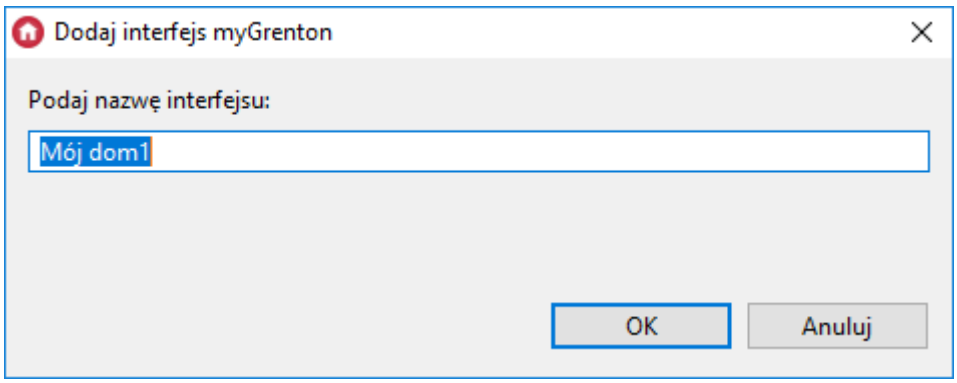
Interfejs DEMO po dodaniu nowego interfejsu zostanie automatycznie usunięty. Możliwe jest usunięcie interfejsu DEMO w standardowy sposób - w oknie zarządzania interfejsem.

2. Tworzenie interfejsu

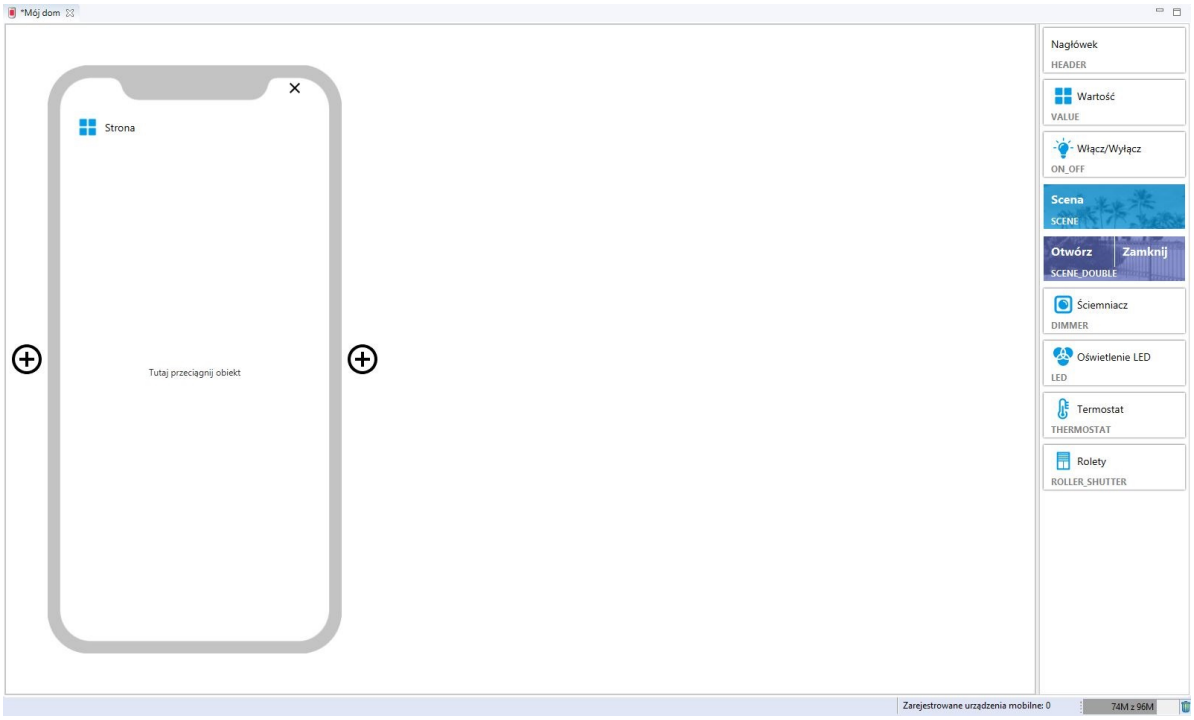
Aby utworzyć interfejs do aplikacji myGrenton należy kliknąć na ikonę `Dodaj interfejs myGrenton` w Menu Głównym Object Managera:



Następnie pojawi się okno z możliwością zmiany nazwy interfejsu:



Po zaakceptowaniu nazwy zostanie utworzony pusty interfejs:



2.1. Dodawanie strony do interfejsu

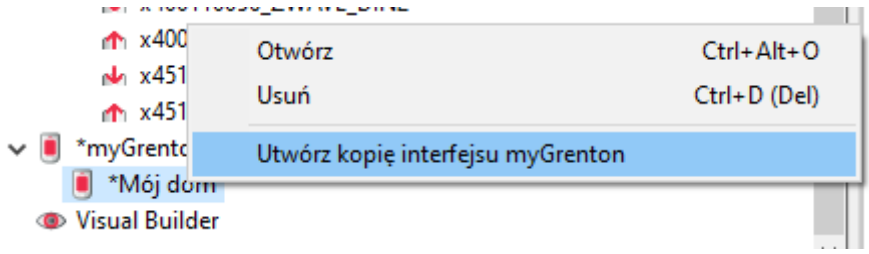
Aby dodać kolejną stronę interfejsu należy kliknąć ikonę `+`, która znajduje się obok grafiki telefonu. Interfejs może posiadać maksymalnie 30 stron.

2.2. Usuwanie strony z interfejsu

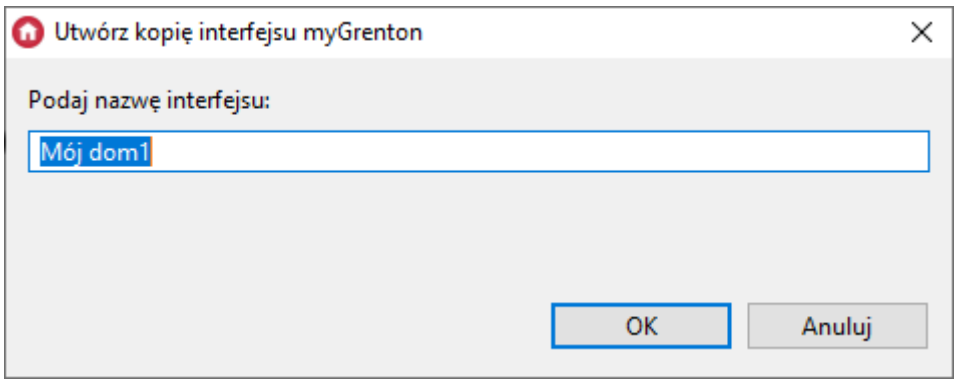
Aby usunąć stronę z interfejsu należy kliknąć ikonę `x`, która znajduje się w prawym górnym rogu grafiki telefonu, a następnie potwierdzić usunięcie strony.

2.3. Kopiowanie interfejsu

Aby skopiować interfejs należy kliknąć PPM na interfejs i wybrać z menu kontekstowego "Utwórz kopię interfejsu myGrenton":



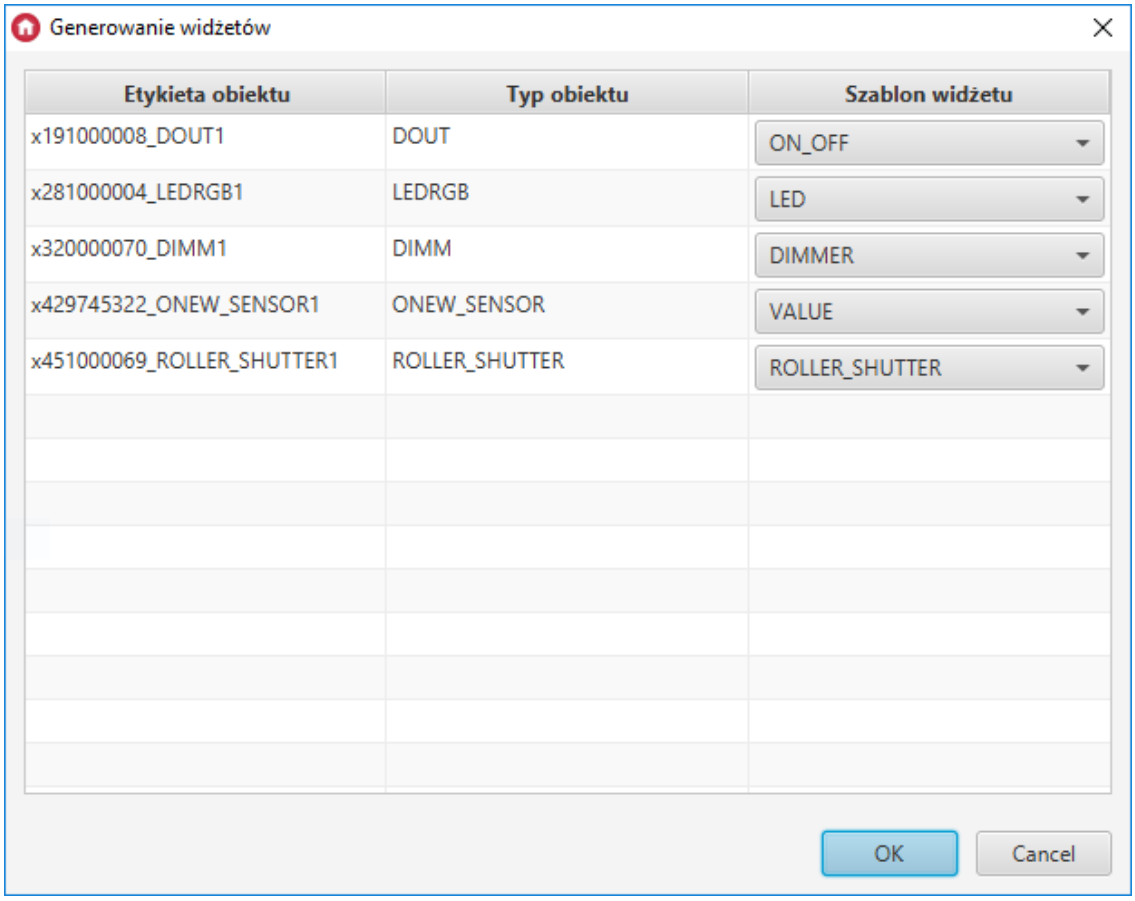
Następnie należy wybrać nazwę dla nowego interfejsu:



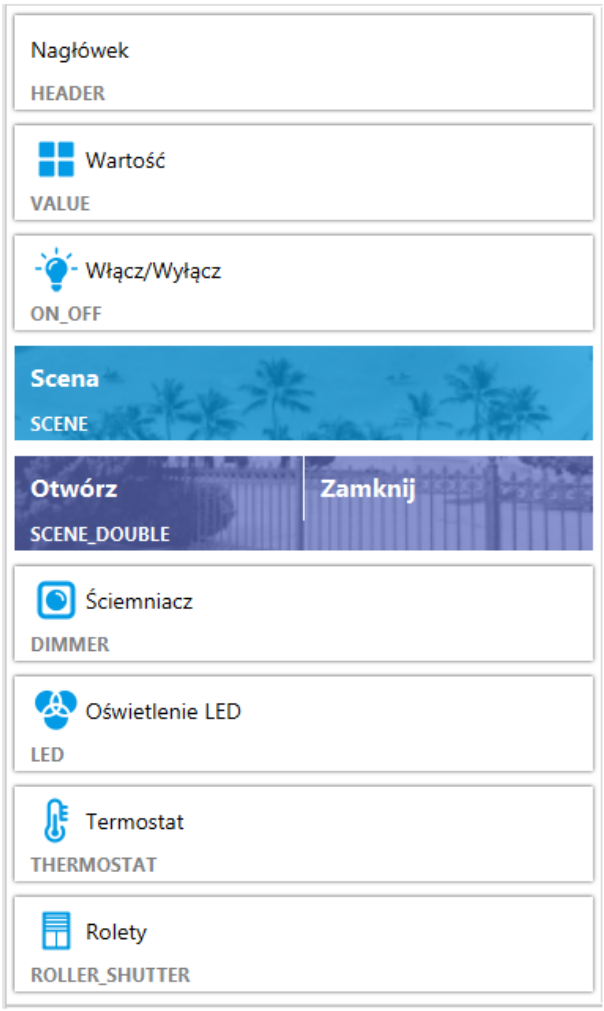
3. Widgety

Do sterowania systemem za pomocą aplikacji myGrenton z poziomu telefonu lub tabletu służą widgety. Każdy z nich oferuje inną funkcjonalność. Widgety mogą zostać dodane do interfejsu na 2 sposoby:

1. Przeciągając konkretny obiekt z listy obiektów w Object Managerze (wtedy utworzony widget będzie posiadał uprzednio zdefiniowany szablon)



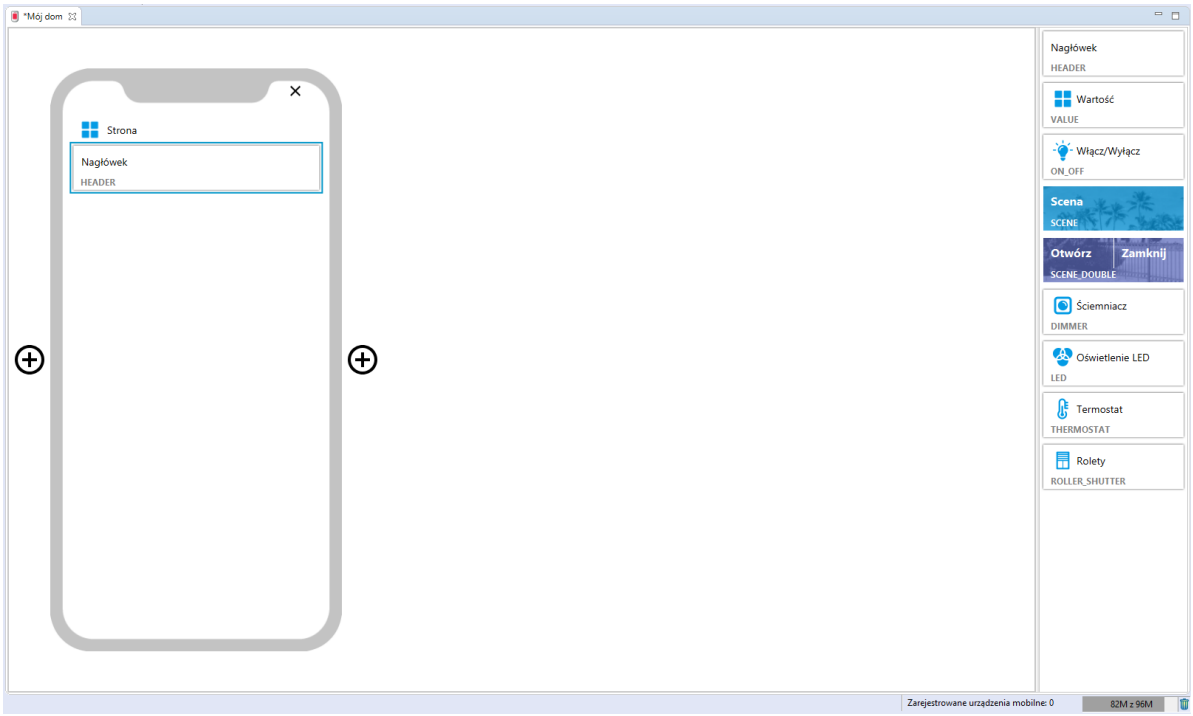
2. Przeciągając konkretny widget z zakładki z prawej strony a następnie uzupełniając go o pożądane parametry.



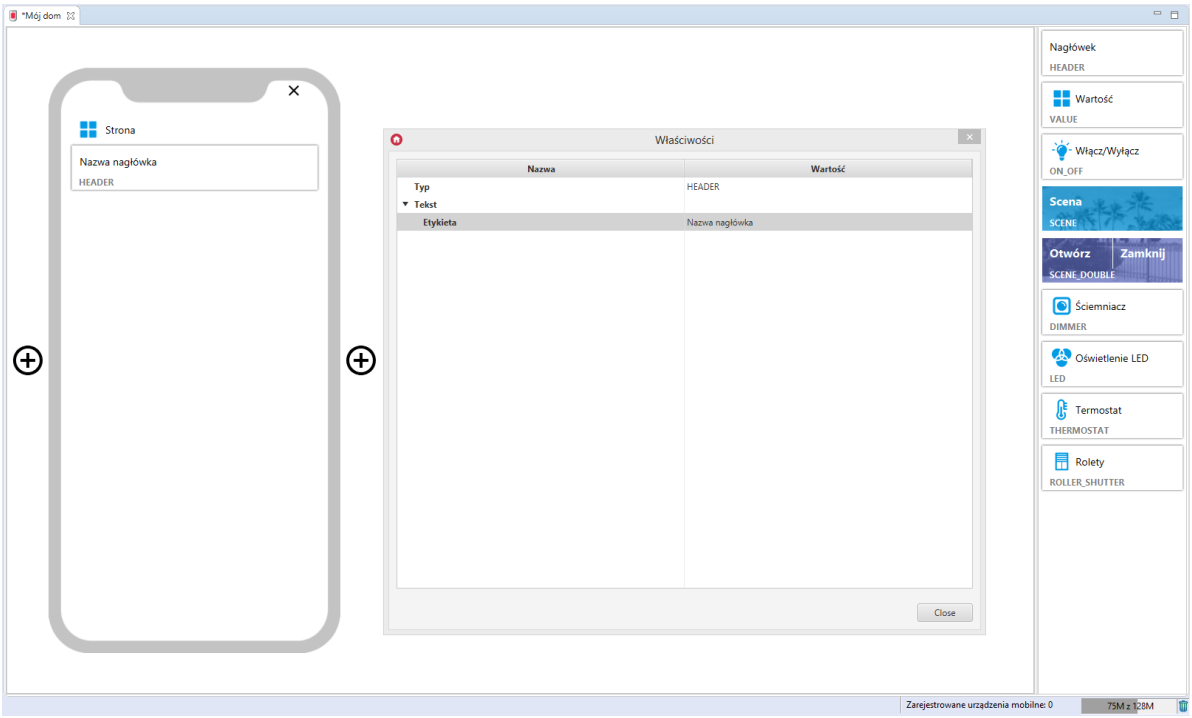
Uwaga! Maksymalna ilość widgetów na stronę wynosi 30.

3.1. Nagłówek (HEADER)

Służy do wyświetlania ciągu znaków. Obsługiwane są wszystkie znaki alfabetyczne, numeryczne oraz znaki specjalne.



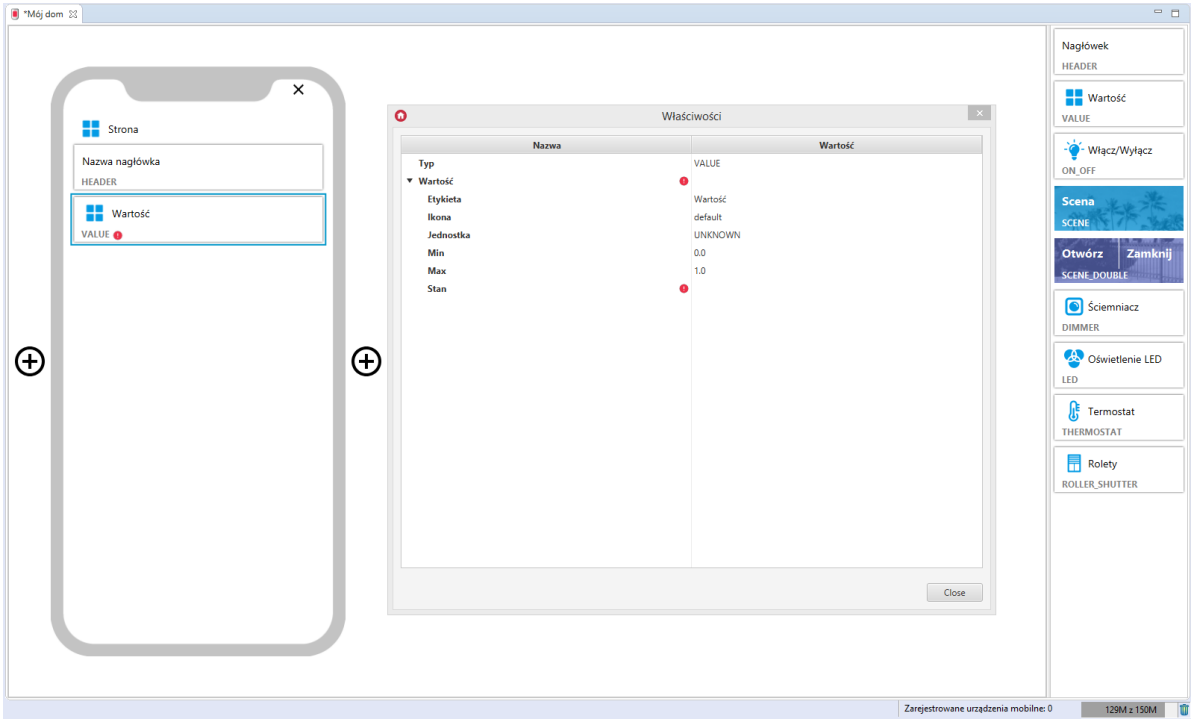
Aby zmienić nagłówek należy kliknąć napis `Nagłówek` w polu telefonu. Możliwa jest również edycja w oknie właściwości w polu `Etykieta`, które pojawia się po dwukrotnym naciśnięciu na widget lewym przyciskiem myszy.



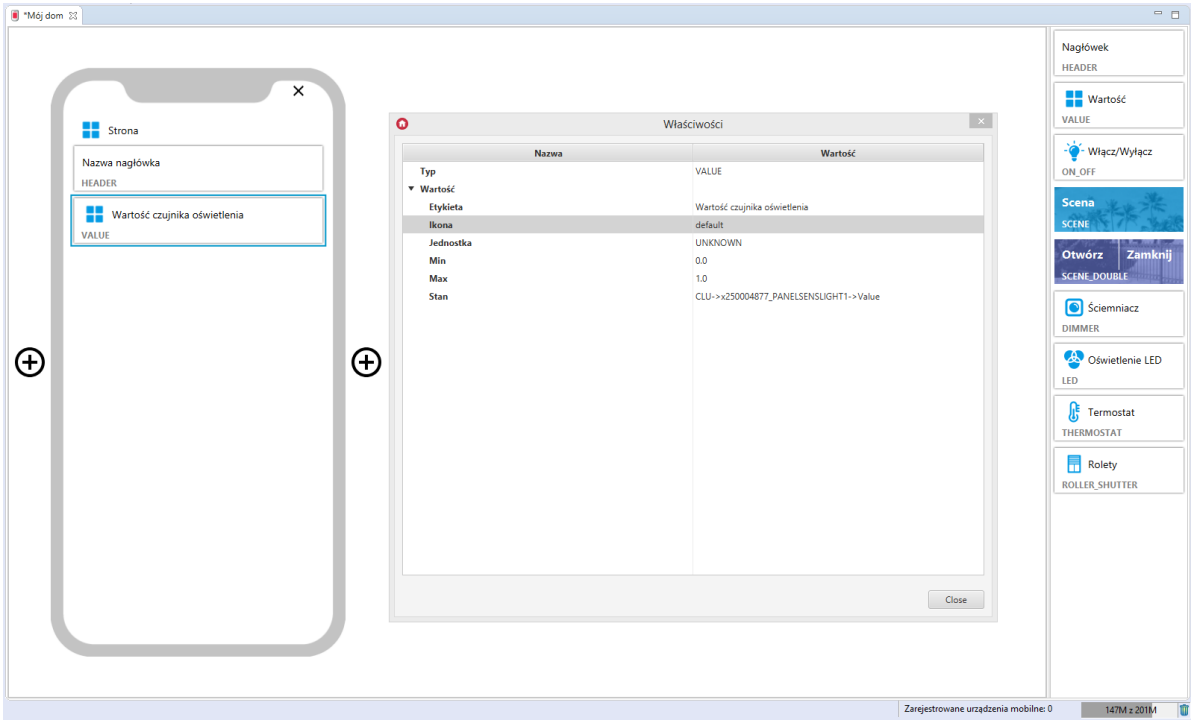
3.2. Wartość (VALUE)

> **Uwaga!** Od wersji Object Manager 1.6.0 widget VALUE oraz możliwość wykorzystania go jako gotowego szablonu będą niedostępne. Zostaje on zastąpiony widgetem VALUE_V2.
>
> Widgety VALUE znajdujące się w projektach stworzonych na poprzednich wersjach Object Manager będą nadal poprawnie obsługiwane i wyświetlane zarówno w Object Mangerze jak i w aplikacji myGrenton.

Widget zwraca wartości cechy danego obiektu. Widget ten dysponuje trzema jednostkami do wyboru: niezdefiniowana (UNKNOWN), procentowa (PERCENT) oraz stopniowa (DEGREE). Ponadto istnieje możliwość opisania widgetu w polu `Etykieta` a także zmiany ikony w polu `Ikona` w oknie właściwości, które pojawia się po dwukrotnym naciśnięciu na widget lewym przyciskiem myszy.



Skonfigurowany widget VALUE:



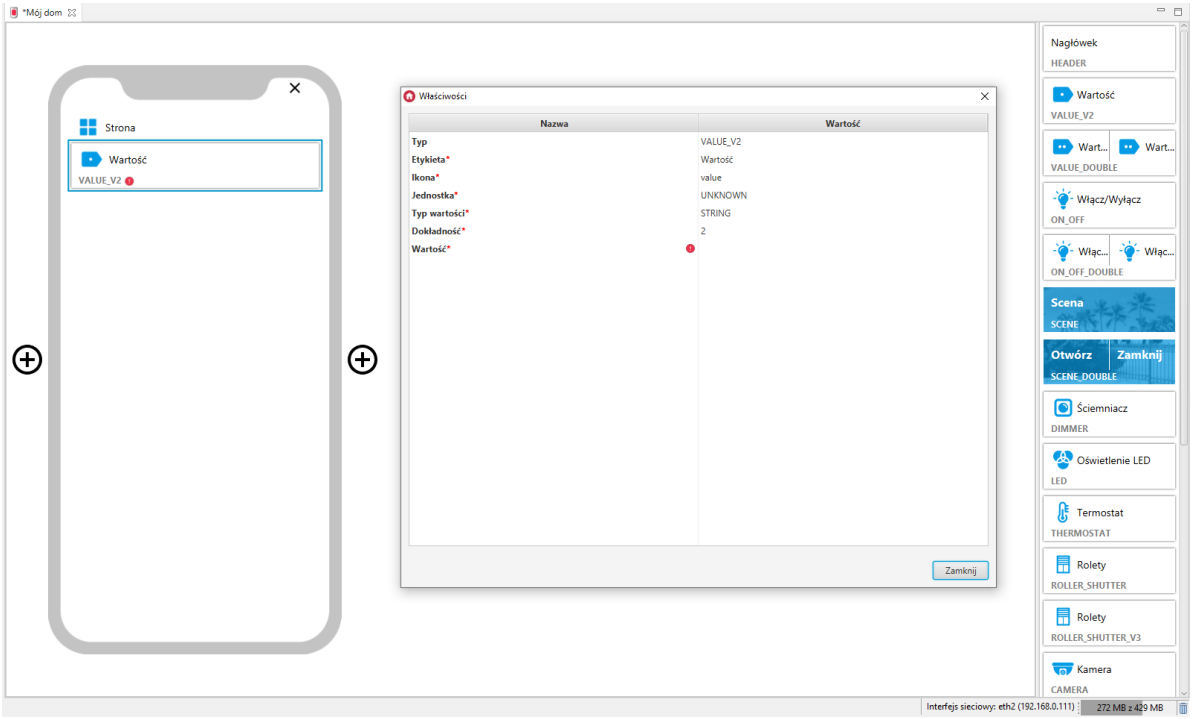
3.3. Wartość v2 (VALUE_V2)

> **Uwaga!** Widget VALUE_V2 dostępny jest dla Object Manager w wersji 1.6.0 lub wyższej, dla aplikacji myGrenton w wersji 1.4.0 (Android) / 1.8.0 (iOS) lub wyższej.

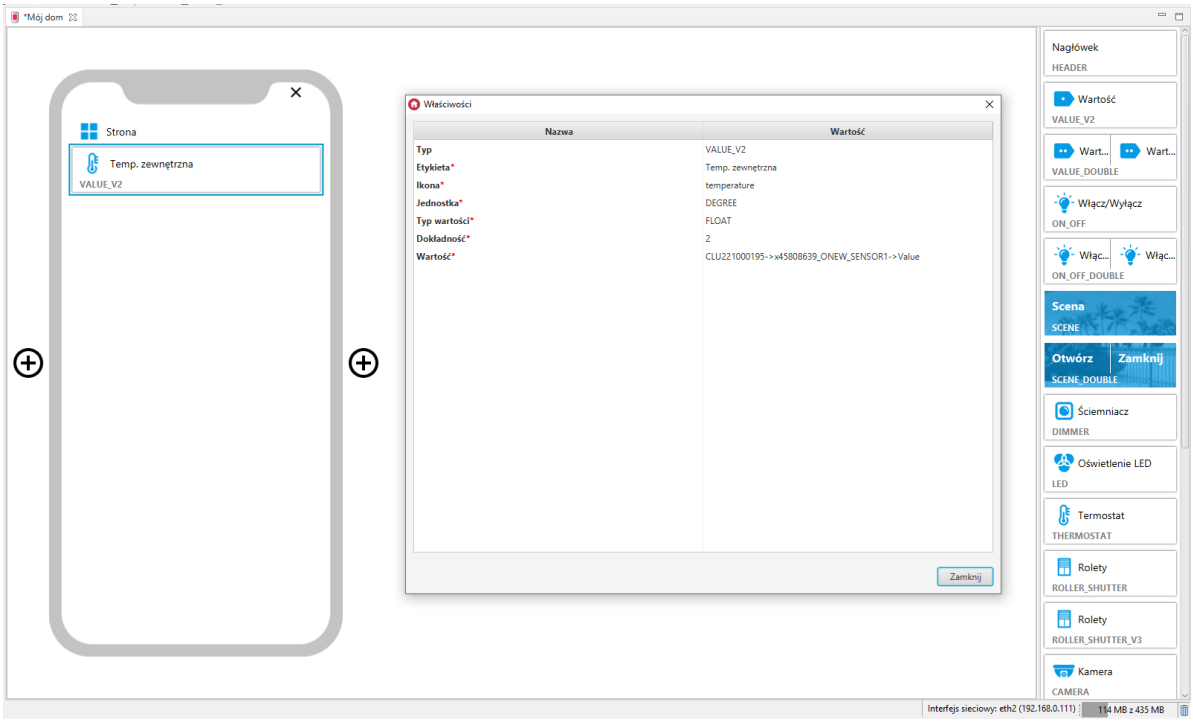
Widget dedykowany do wyświetlania wartości cechy użytkownika lub cechy wbudowanej danego obiektu. Dysponuje trzema jednostkami do wyboru: niezdefiniowaną (UNKNOWN), procentową (PERCENT) oraz stopniową (DEGREE). Wybrana jednostka jest wyświetlana obok wartości. Ponadto należy wybrać typ:

- STRING - tekstowy,
- FLOAT - zmiennoprzecinkowy, dla tego typu wartości możliwe określenie ilości wyświetlanych miejsc po przecinku (właściwość `Dokładność`),

- INTEGER - całkowity.



Skonfigurowany widget VALUE_V2:



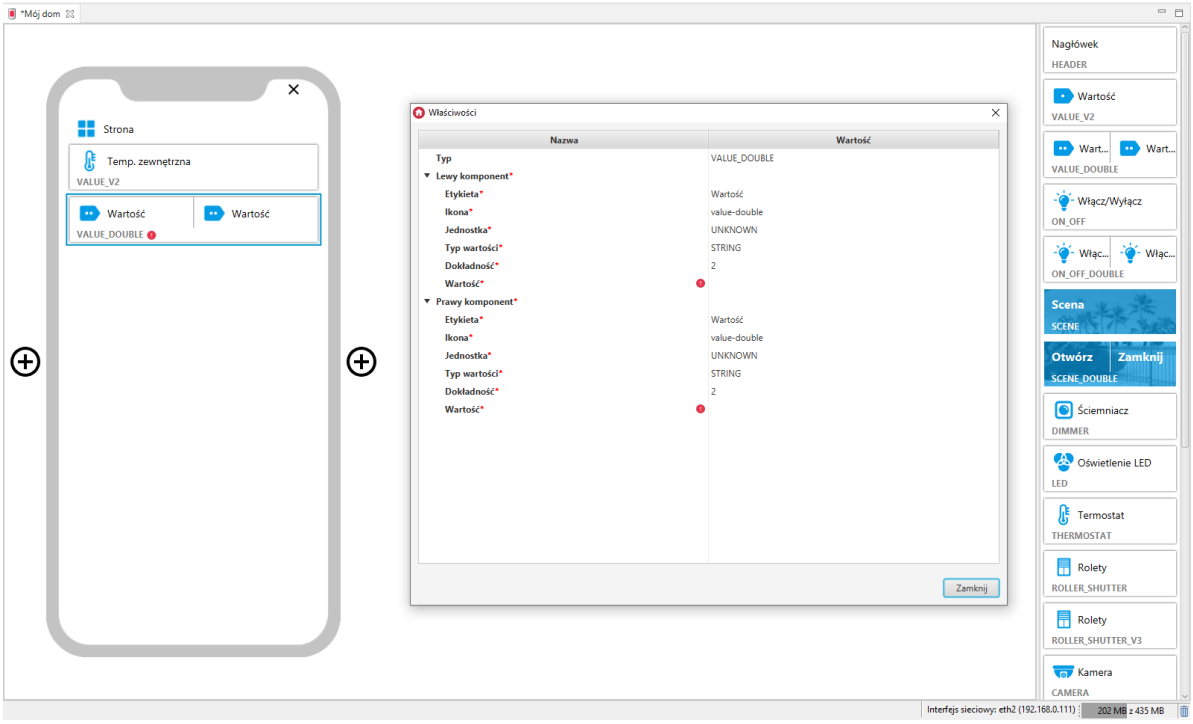
3.4. Wartość Podwójny (VALUE_DOUBLE)

> **Uwaga!** Widget VALUE_DOUBLE dostępny jest dla Object Manager w wersji 1.6.0 lub wyższej, dla aplikacji myGrenton w wersji 1.4.0 (Android) / 1.8.0 (iOS) lub wyższej.

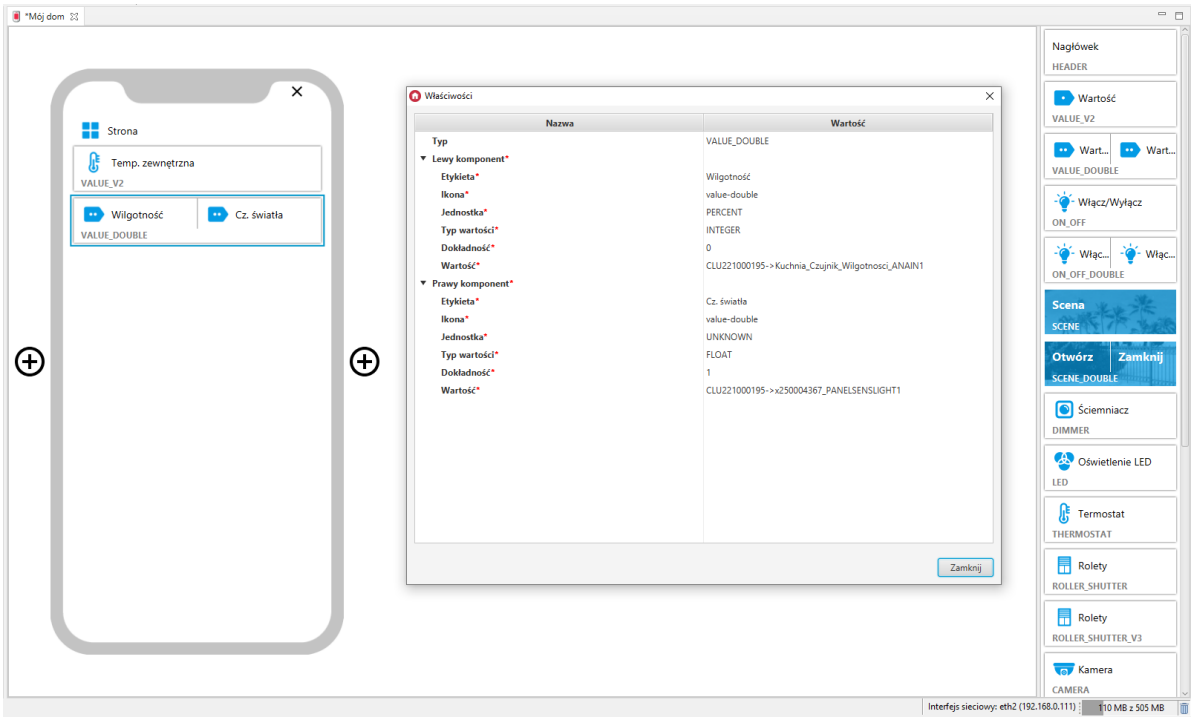
Jest to podwójna wersja widgetu Wartość v2. Widget dedykowany do wyświetlania wartości cech użytkownika lub cech wbudowanych danego obiektu. Dysponuje trzema jednostkami do wyboru: niezdefiniowaną (UNKNOWN), procentową (PERCENT) oraz stopniową (DEGREE). Wybrana jednostka jest wyświetlana obok wartości. Ponadto należy wybrać typ:

- STRING - tekstowy,

- FLOAT - zmiennoprzecinkowy, dla tego typu wartości możliwe określenie ilości wyświetlanych miejsc po przecinku (właściwość `Dokładność`),
- INTEGER - całkowity.

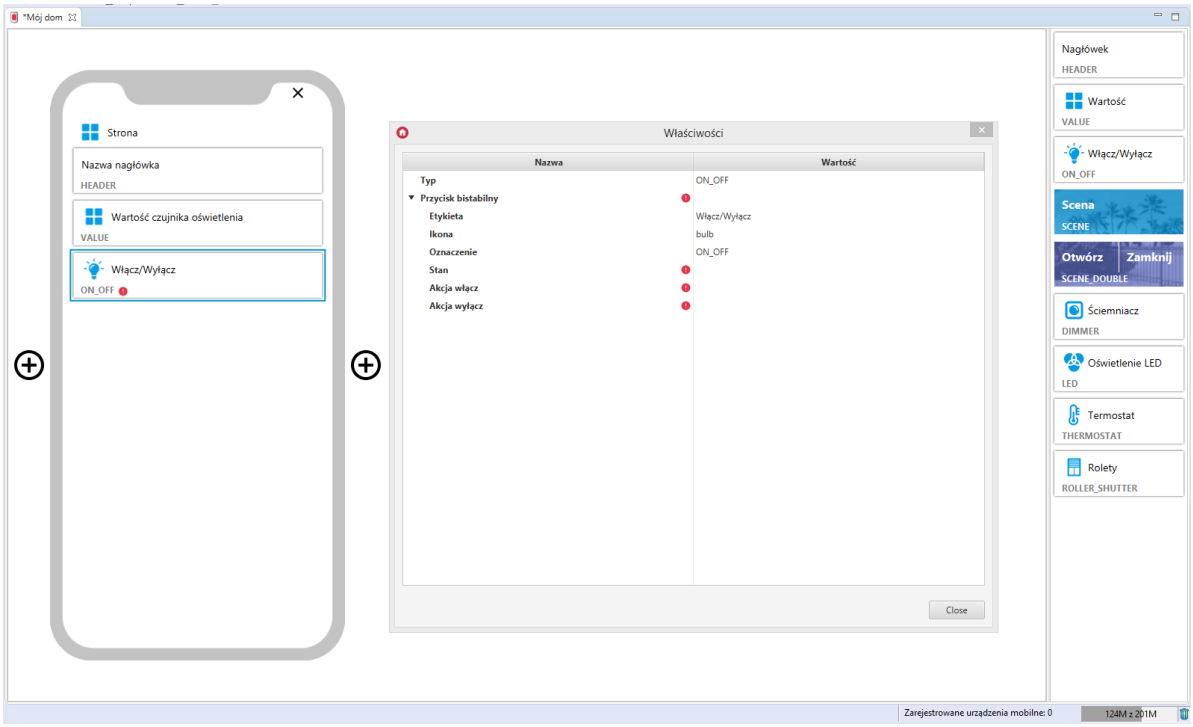


Skonfigurowany widget VALUE_DOUBLE:

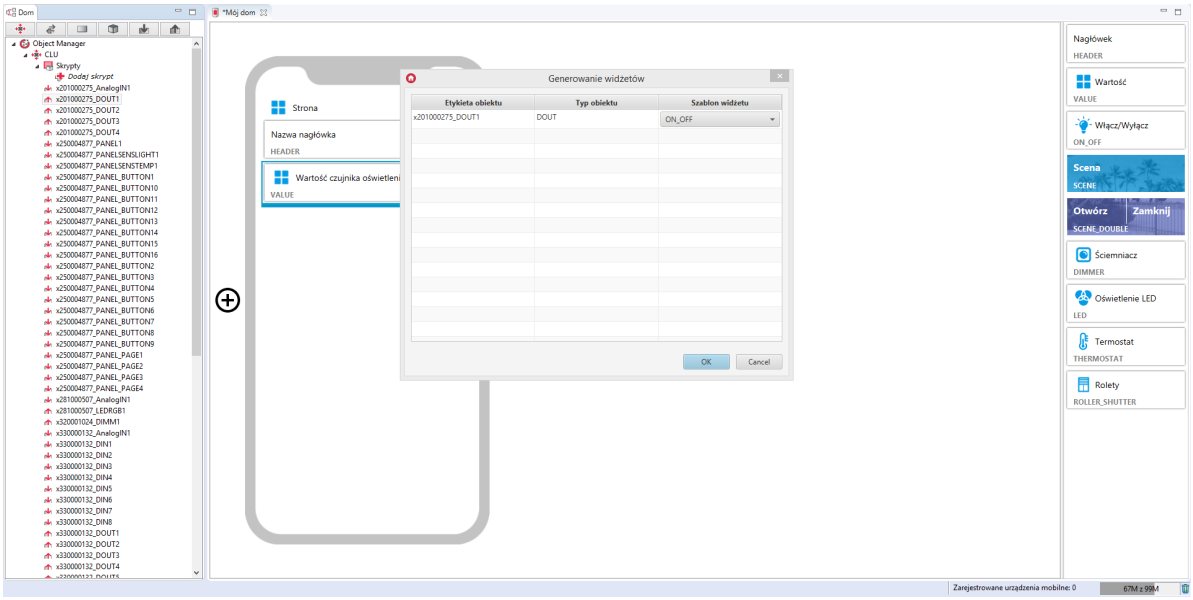


3.5. Włącz/Wyłącz (ON_OFF)

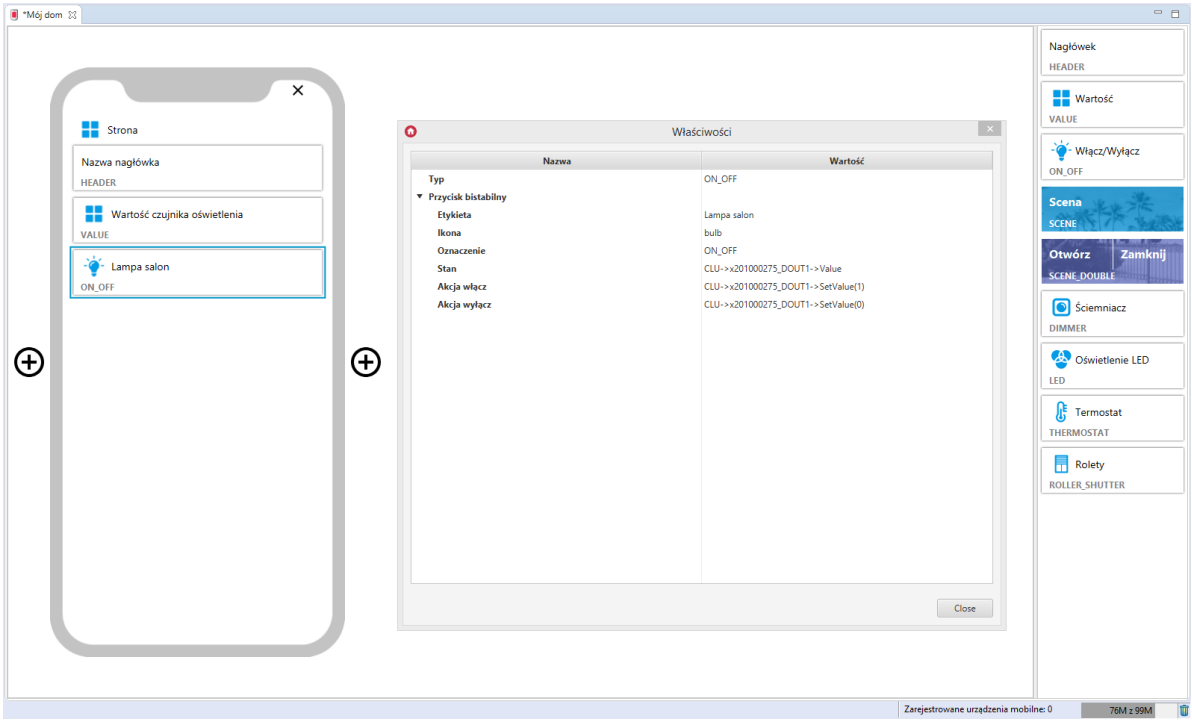
Widget służy do sterowania elementami wykonawczymi. Przeciągając widget z zakładki z prawej strony ekranu należy go uzupełnić o wartości w polach `Stan`, `Akcja włącz`, `Akcja wyłącz`. Widget dedykowany głównie do sterowania wyjściami przekaźnikowymi, jednak może być zastosowany np. do załączania/wyłączania oświetlenia LED lub do uruchamiania obiektów wirtualnych.



Dla obiektów wyjść cyfrowych zdefiniowane są gotowe szablony dla widgetu ON_OFF. Aby dodać widget ON_OFF z pożądanym obiektem wyjścia cyfrowego należy przeciągnąć obiekt DOUT z listy obiektów na stronę interfejsu:



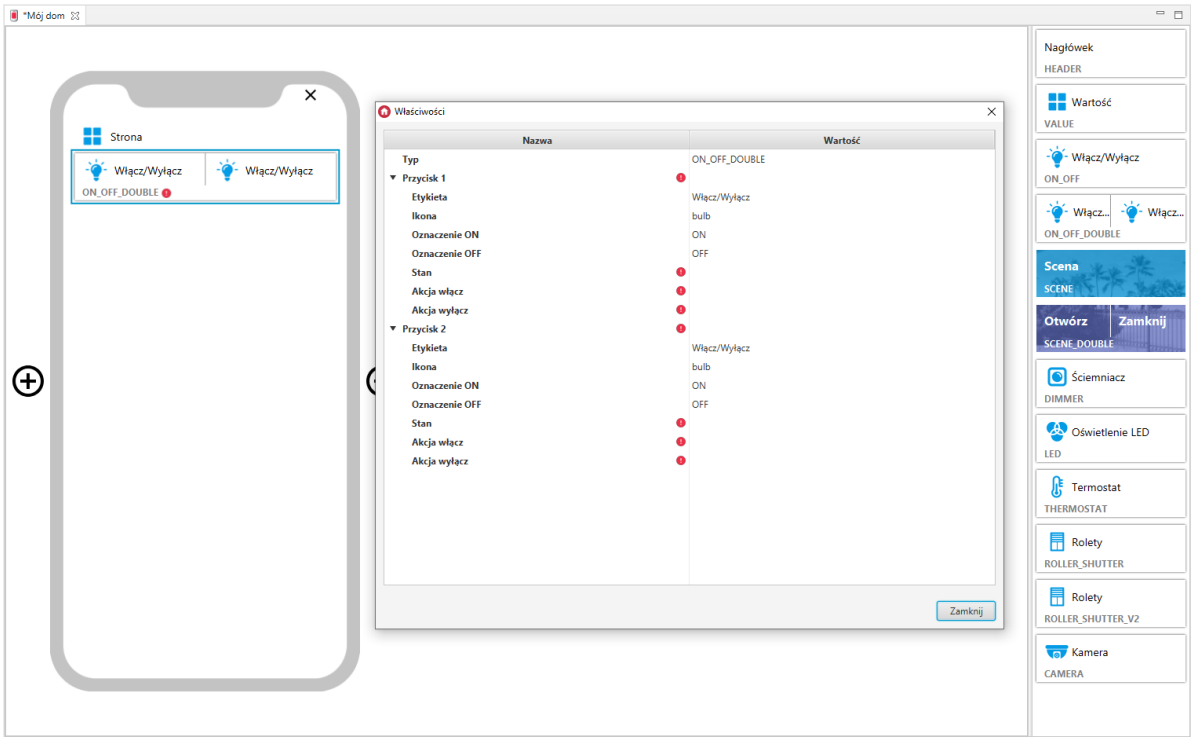
Skonfigurowany widget ON_OFF:



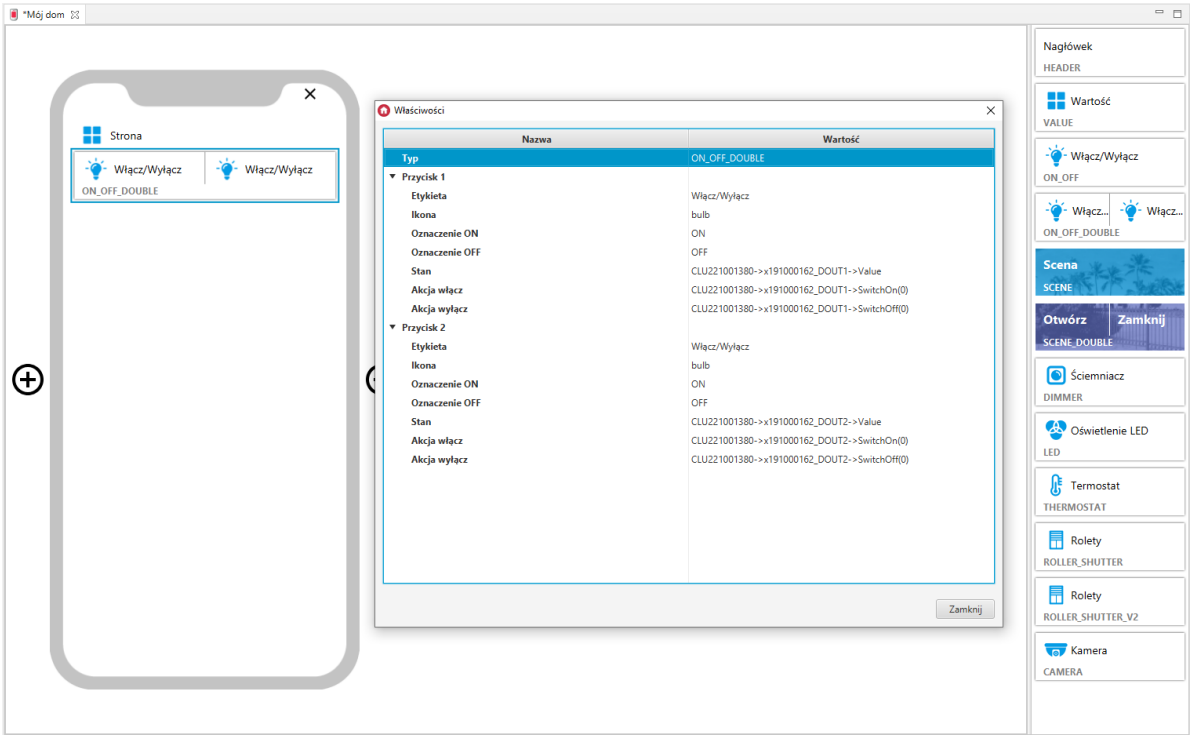
3.6. Włącz/Wyłącz Podwójny (ON_OFF_DOUBLE)

> **Uwaga!** Widget ON_OFF_DOUBLE dostępny jest dla Object Manager w wersji 1.4.0 lub wyższej, oraz dla aplikacji myGrenton w wersji 1.2.3 lub wyższej (Android) i wersji 1.6.0 lub wyższej (iOS).

Jest to podwójna wersja widgetu Włącz/Wyłącz. Przeciągając widget z zakładki z prawej strony ekranu należy go uzupełnić o wartości w polach `Stan`, `Akcja włącz`, `Akcja wyłącz` dla obu przycisków.

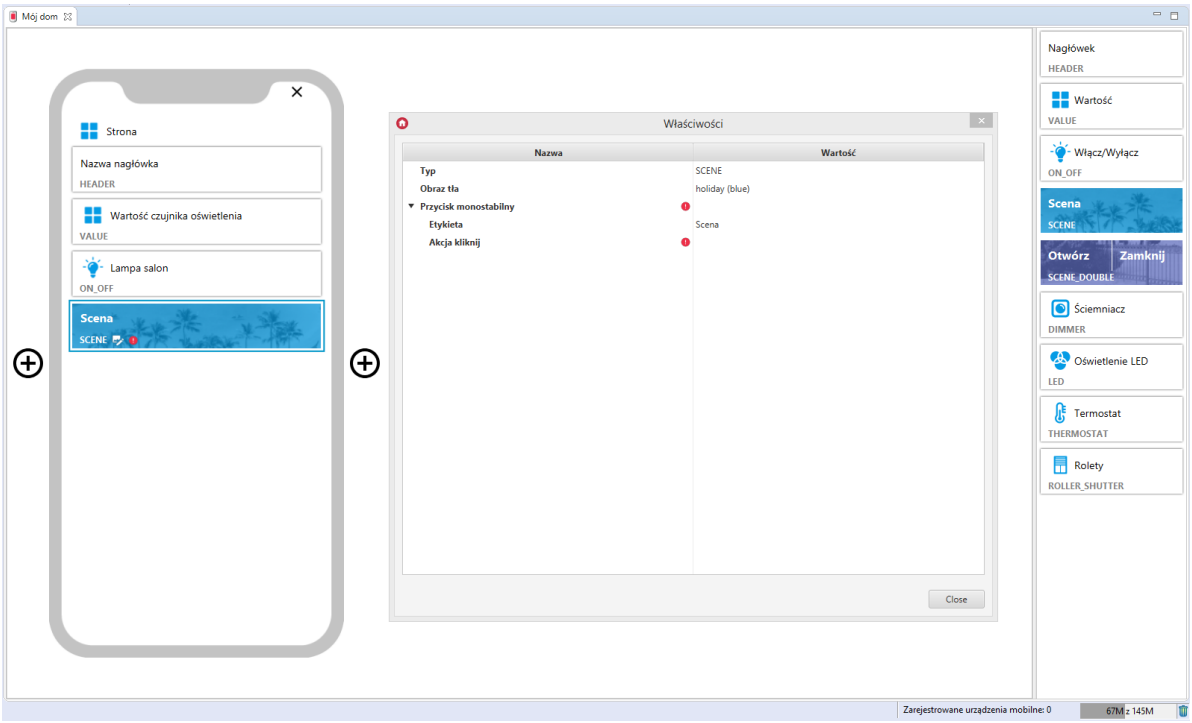


Skonfigurowany widget ON_OFF_DOUBLE:

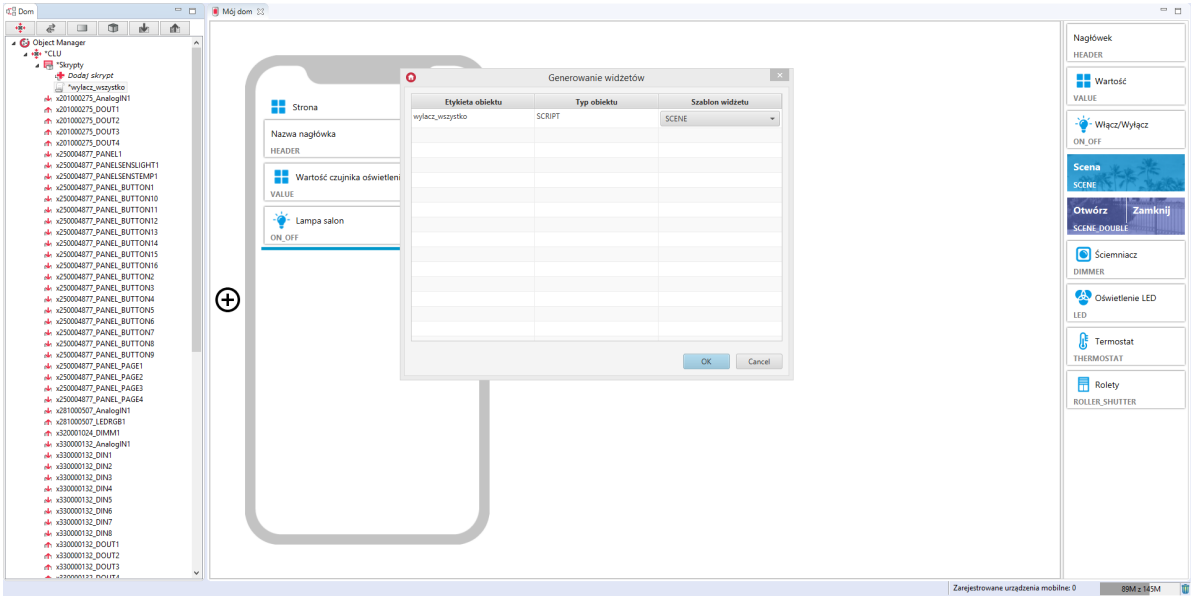


3.7. Scena (SCENE)

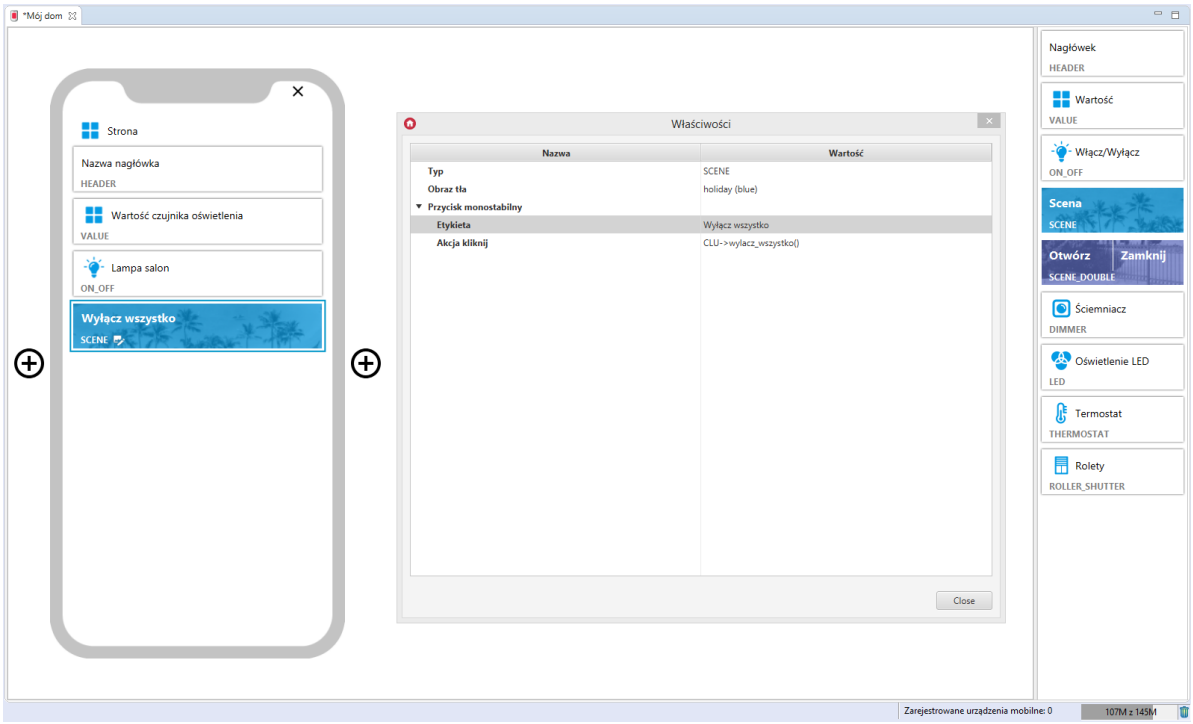
Widget dedykowany jest do wywoływania utworzonych skryptów.



Dla skryptów zdefiniowane są gotowe szablony dla widgetu SCENE. Aby dodać widget SCENE z pożądanym obiektem wyjścia cyfrowego należy przeciągnąć skrypt z listy obiektów na stronę interfejsu:

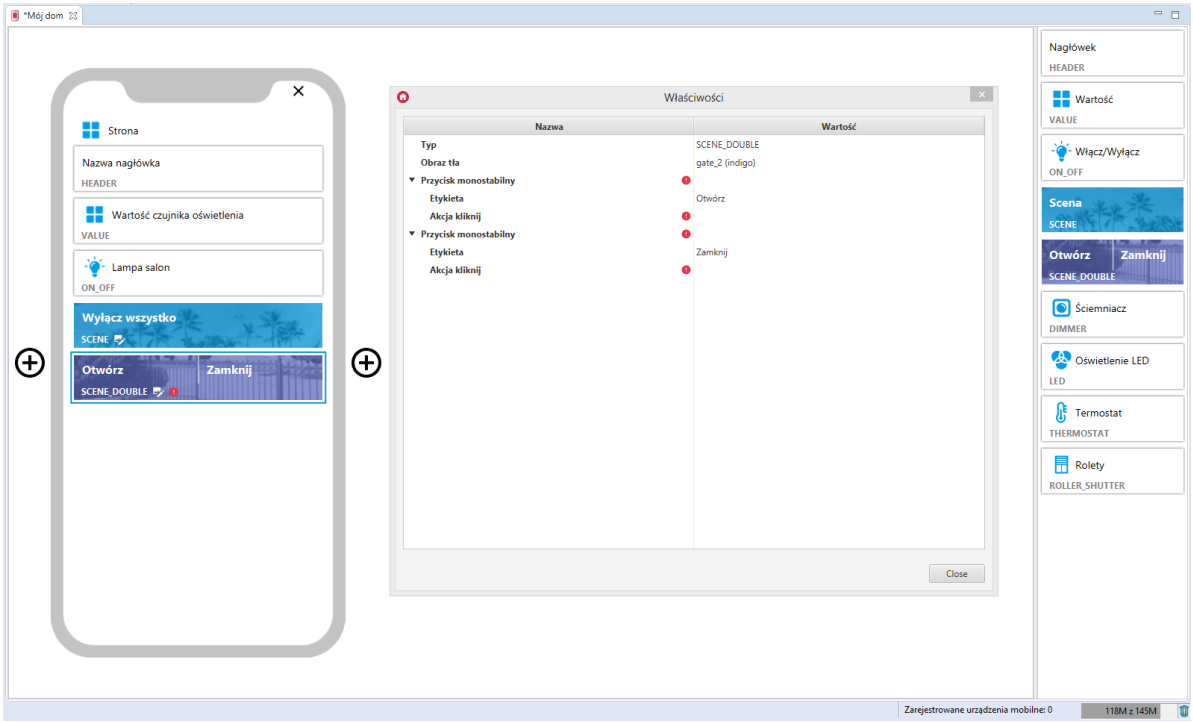


Skonfigurowany widget SCENE:

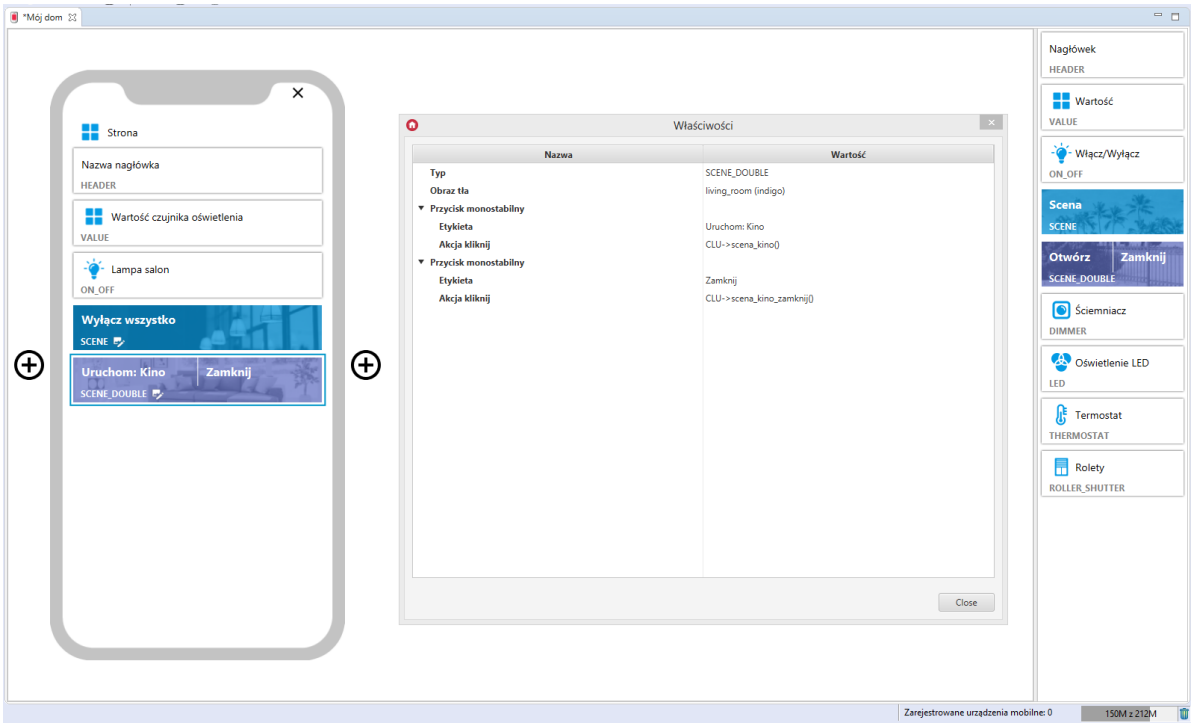


3.8. Scena Podwójny (SCENE_DOUBLE)

Widget służy do wywoływania skryptów (metod obiektów). W odróżnieniu do widgetu SCENE posiada 2 przyciski na akcję.



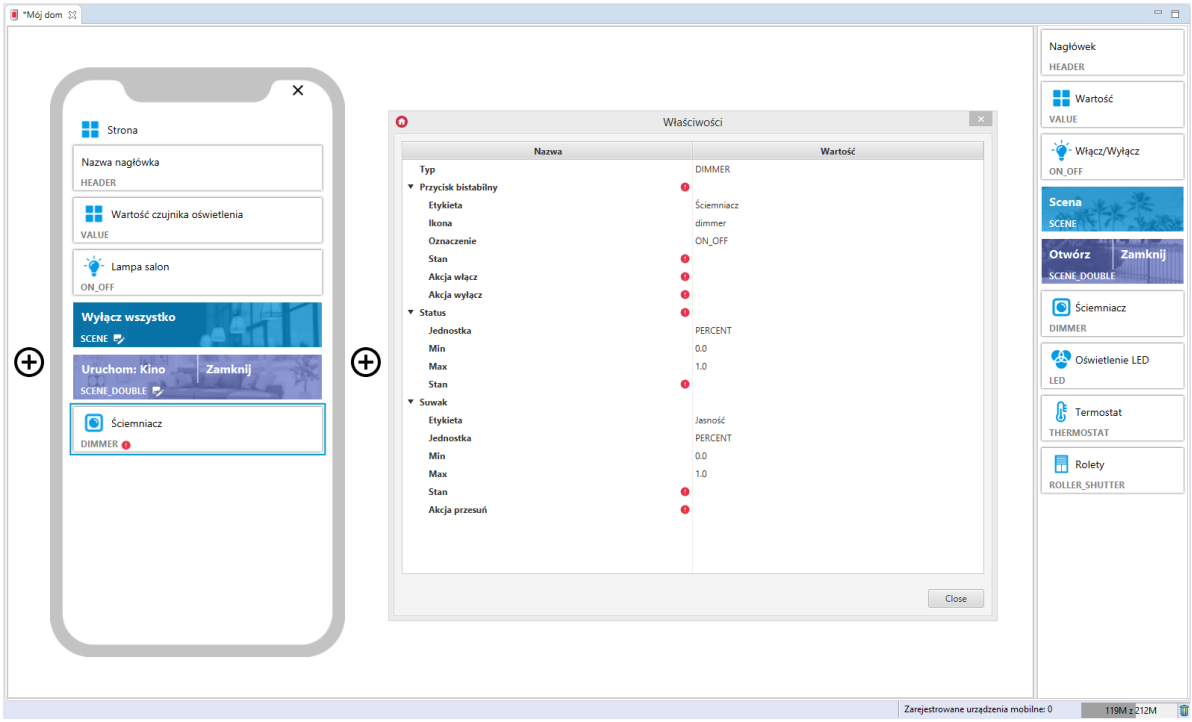
Skonfigurowany widget SCENE_DOUBLE:



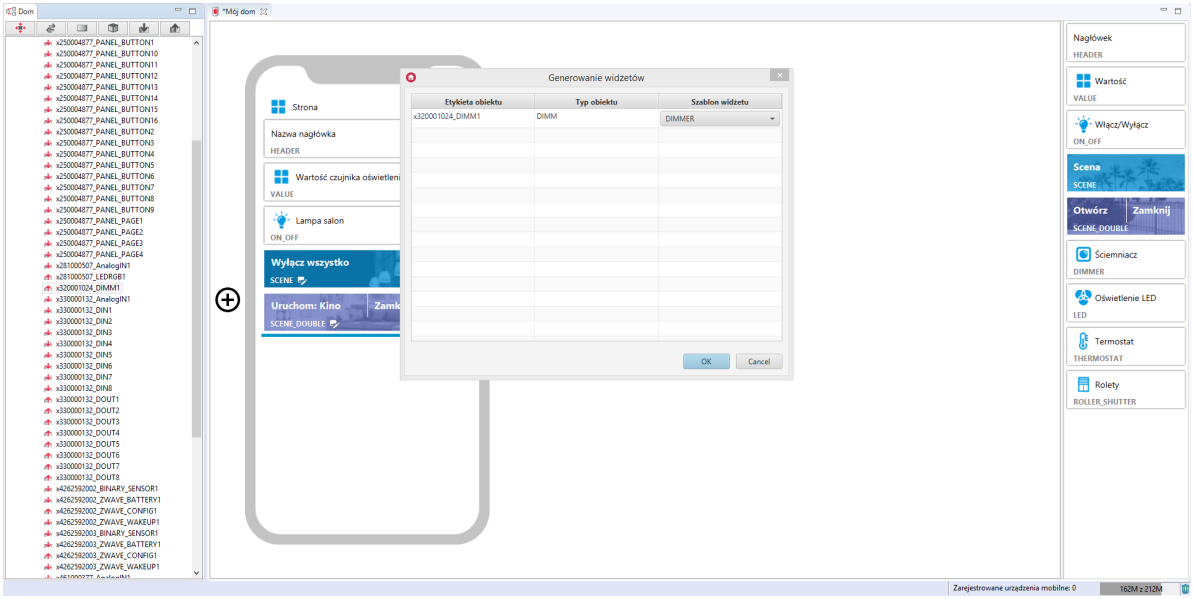
3.9. Ściemniacz (DIMMER)

> **Uwaga!** Od wersji Object Manager 1.7.0 widget DIMMER oraz możliwość wykorzystania go jako gotowego szablonu będą niedostępne. Zostaje on zastąpiony widgetem DIMMER_V2.
>
> Widgety DIMMER znajdujące się w projektach stworzonych na poprzednich wersjach Object Manager będą nadal poprawnie obsługiwane i wyświetlane zarówno w Object Mangerze jak i w aplikacji myGrenton.

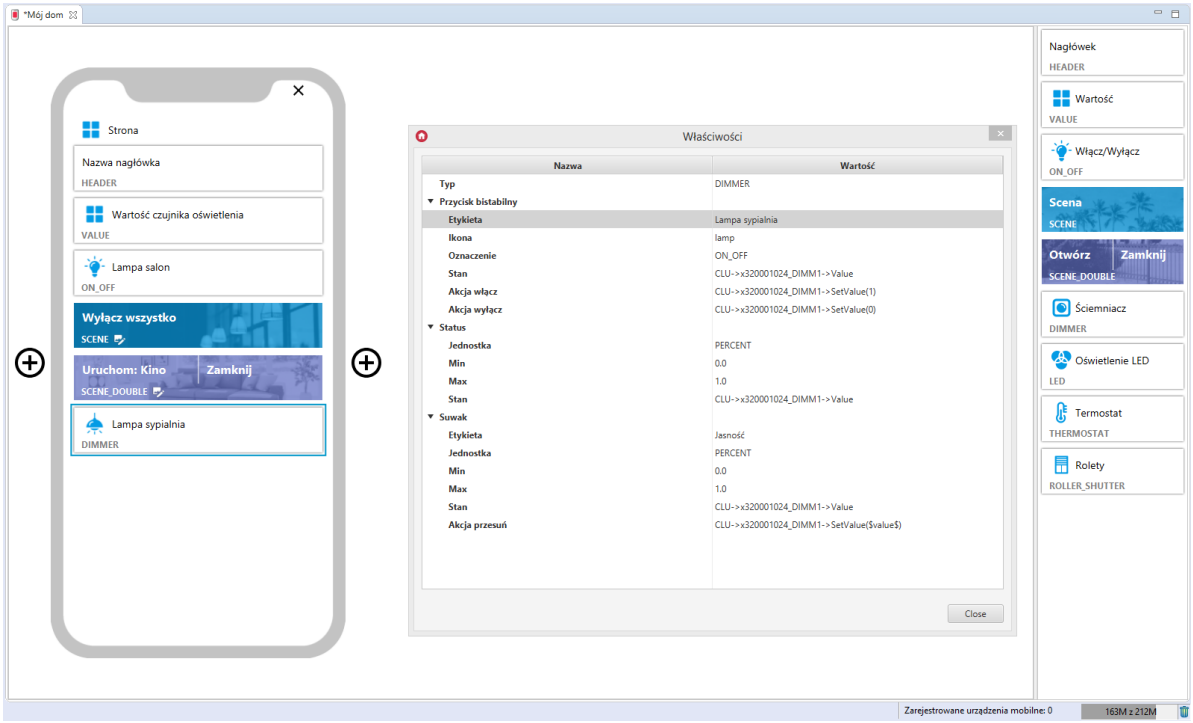
Widget dedykowany dla modułów DIMMER i LEDRGB. Przeciągając widget z zakładki z prawej strony ekranu należy go uzupełnić o wartości w polach `Stan`, `Akcja włącz`, `Akcja wyłącz`, `Jednostka`, `Min`, `Max`, `Akcja przesuń`. Widget ten posiada suwak, dzięki czemu możliwe jest sterowanie w danym zakresie.



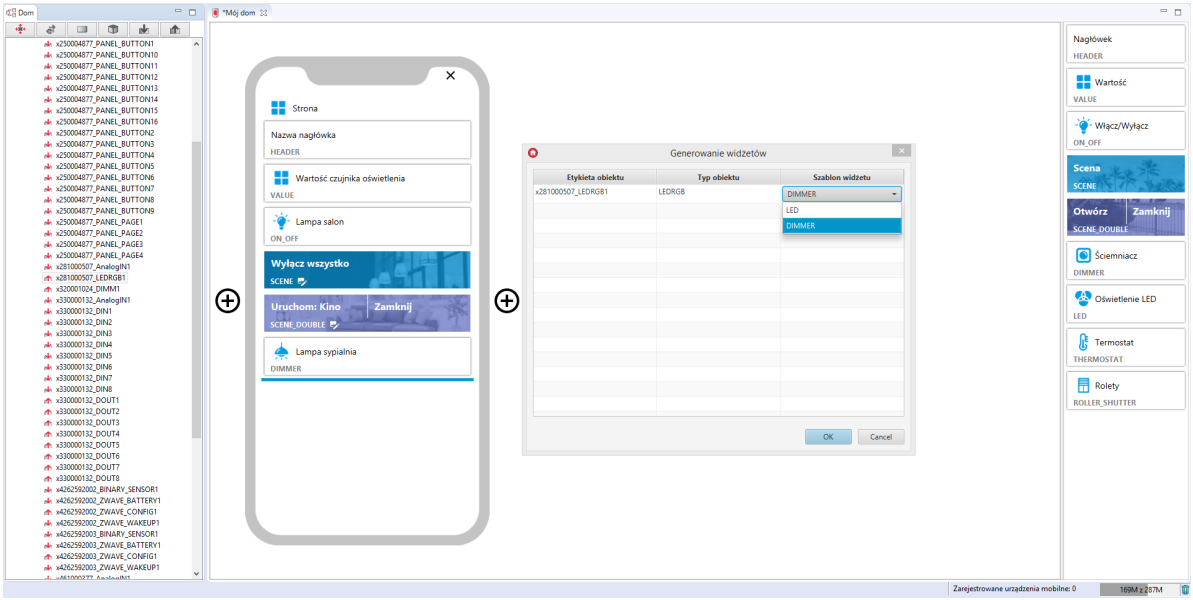
Dla obiektu DIMM zdefiniowane są gotowe szablony dla widgetu DIMMER. Aby dodać widget DIMMER z gotowym szablonem należy przeciągnąć obiekt DIMM z listy obiektów na stronę interfejsu:



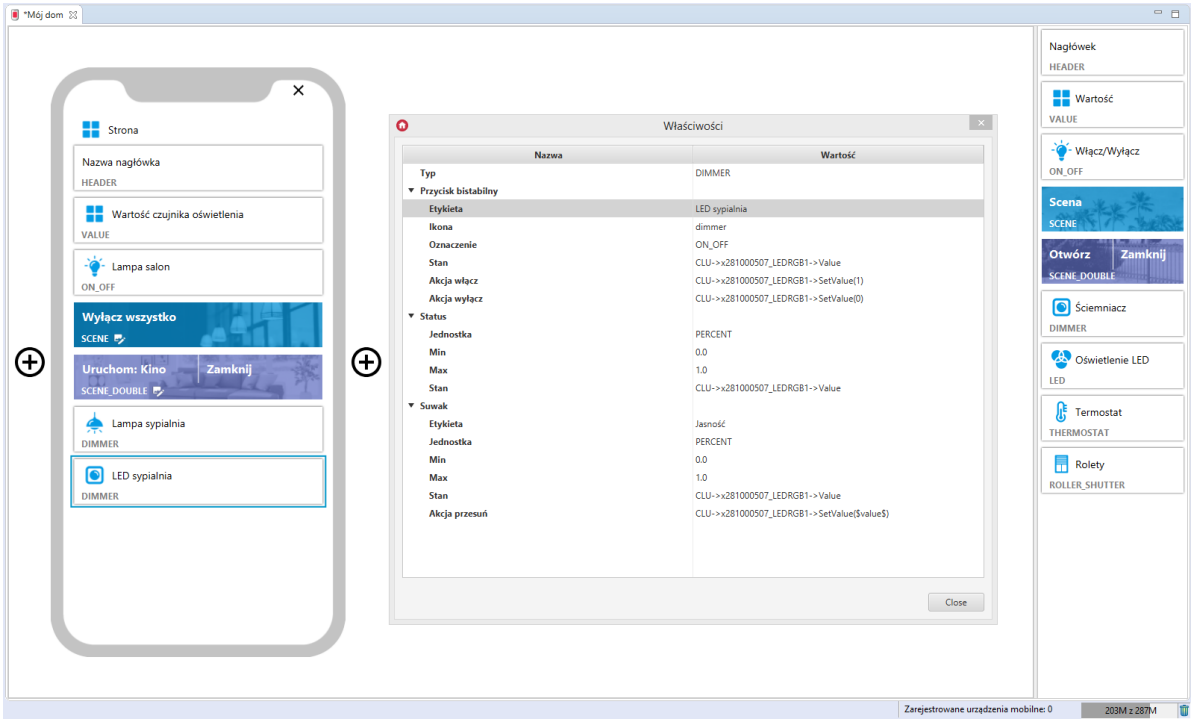
Skonfigurowany widget DIMMER:



Widget DIMMER posiada gotowy szablon także dla obiektu LEDRGB:



Utworzony widget:

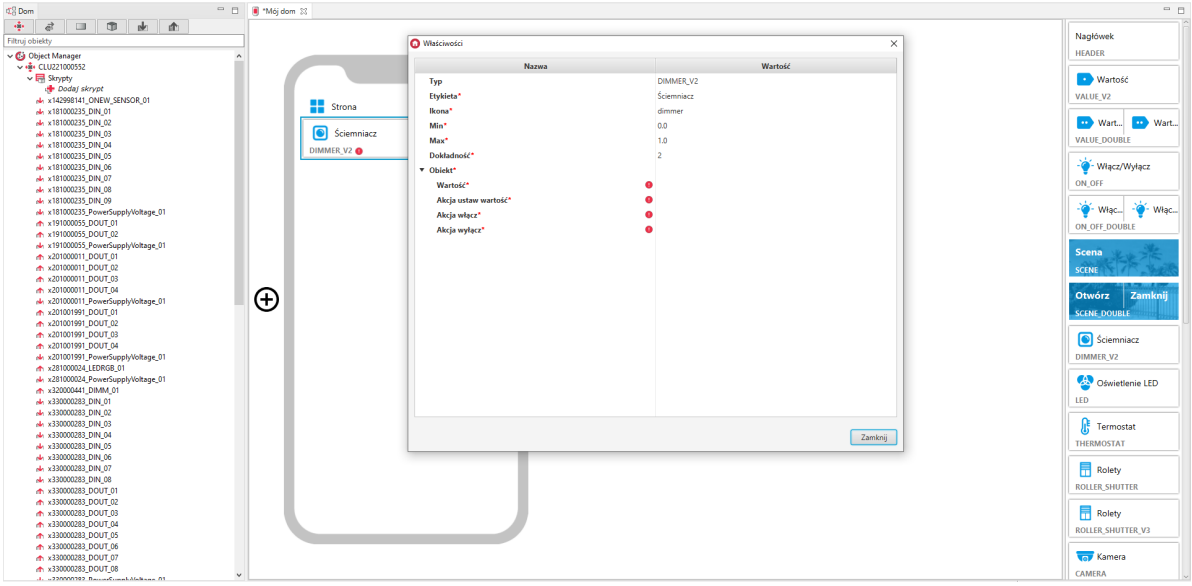


3.10. Ściemniacz v2 (DIMMER_V2)

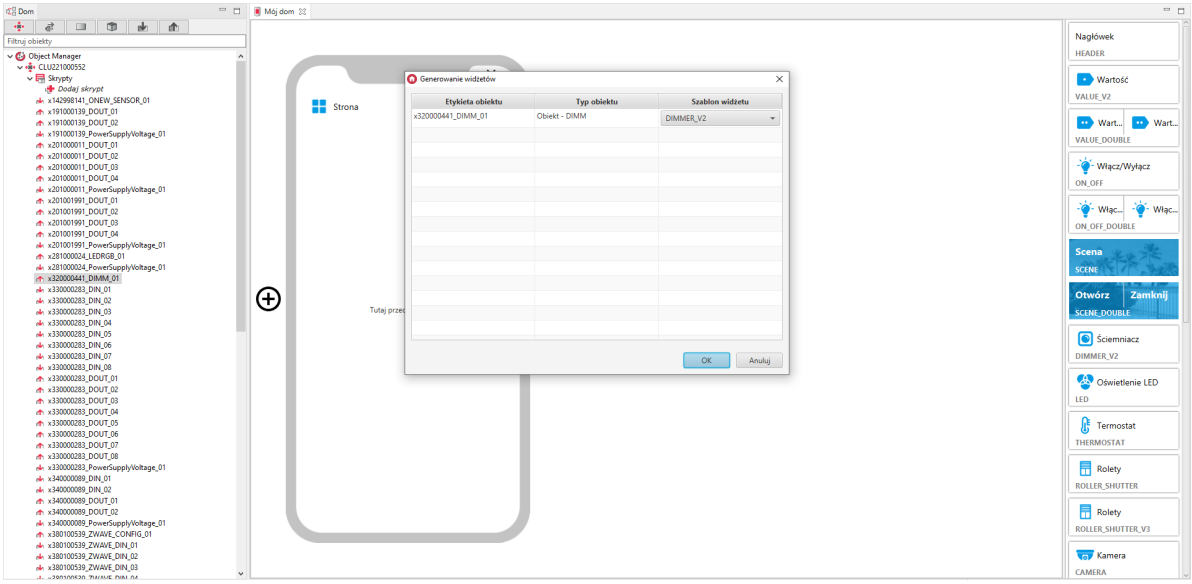
> **Uwaga!** Widget DIMMER_V2 dostępny jest dla Object Manager w wersji 1.7.0 lub wyższej, dla aplikacji myGrenton w wersji 1.5.0 (Android) / 1.9.0 (iOS) lub wyższej.

Widget dedykowany dla modułów DIMMER i LEDRGBW, umożliwiający płynne sterowanie oświetleniem. Widget DIMMER_V2 zawiera:

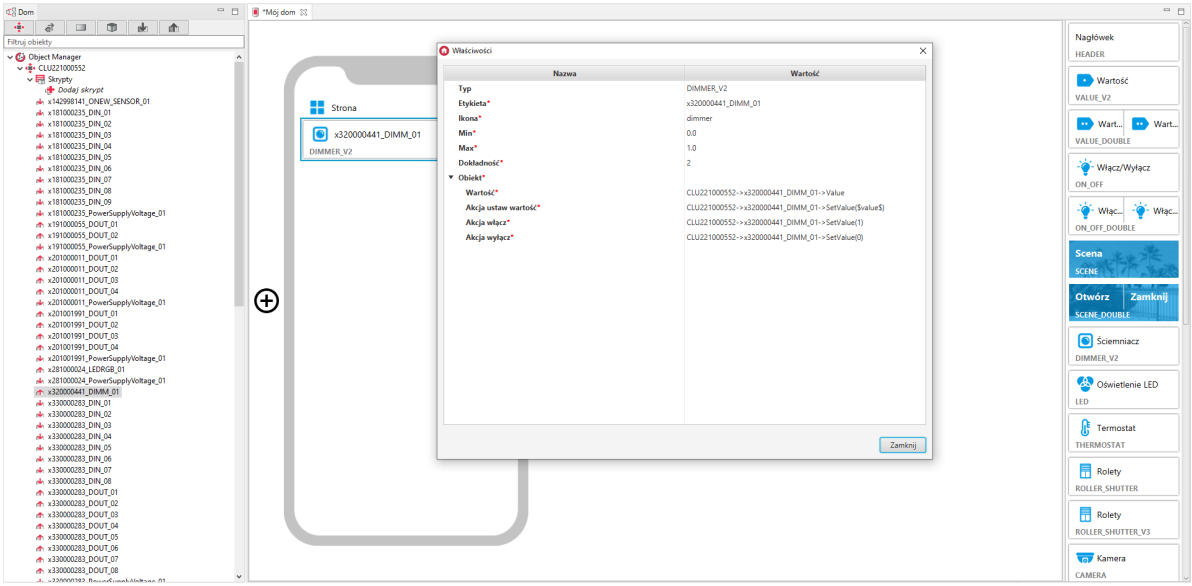
- wartość - aktualna wartość wyjścia wyrażona w procentach, wyświetlana w prawej części widgetu (obliczana na podstawie ustawionych właściwości `Min`, `Max`),
- akcje włącz / wyłącz - akcje wywoływane w momencie kliknięcia na widget załączające / wyłączające wyjście,
- suwak sterujący wyjściem - pracuje w trybie procentowym, wartość wyjścia ustawiana na podstawie podanego zakresu (właściwości `Min`, `Max`) oraz dokładności (właściwość `Dokładność` określa ilość miejsc po przecinku ustawianej wartości).



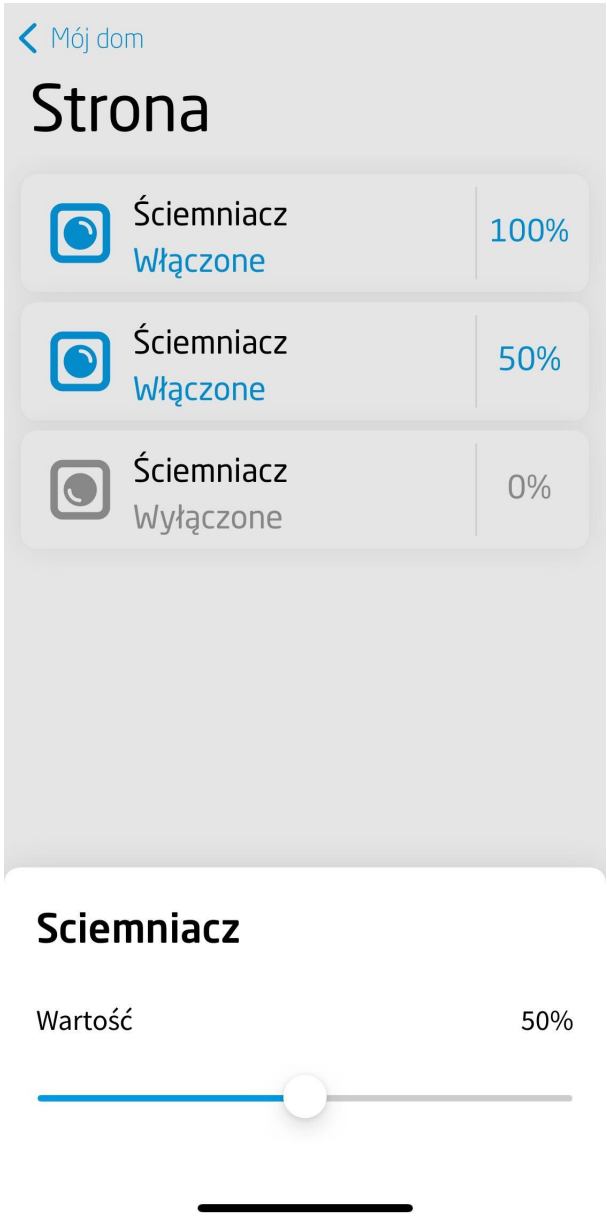
Dla obiektów DIMM, LEDRGB, LEDRGBW zdefiniowane są gotowe szablony dla widgetu DIMMER_V2. Aby dodać widget DIMMER_V2 z gotowym szablonem należy przeciągnąć obiekt z listy obiektów na stronę interfejsu:



Skonfigurowany widget DIMMER_V2:

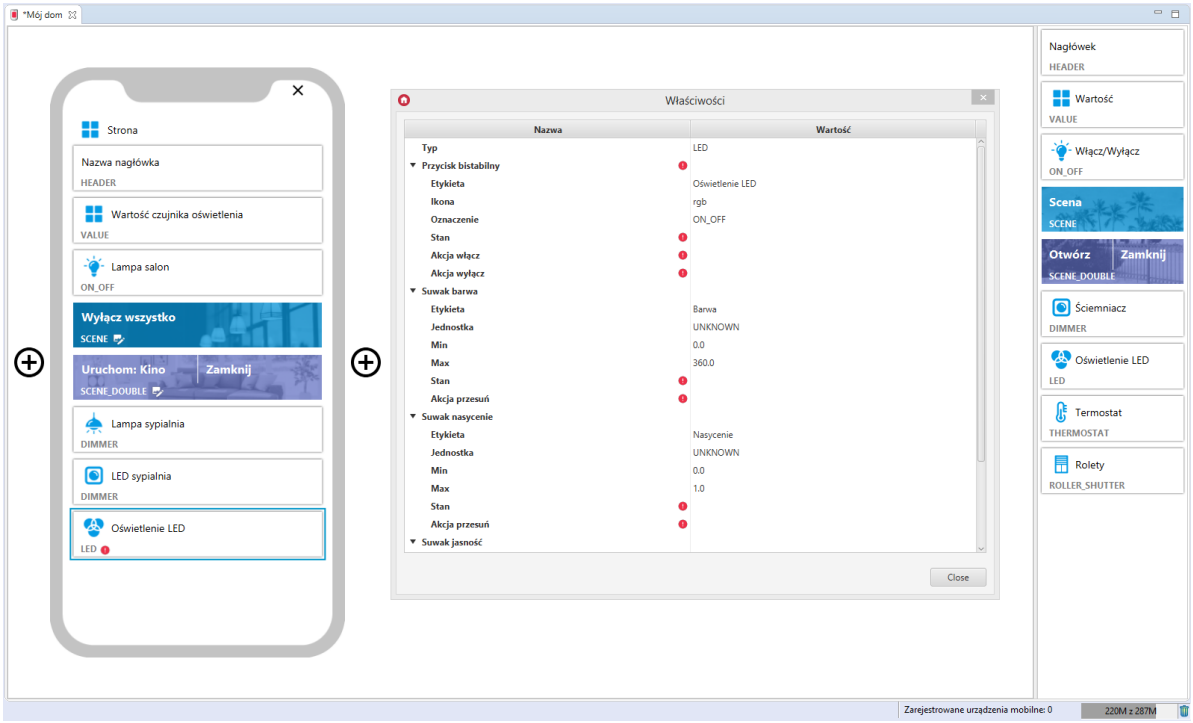


Wygląd widgetu w aplikacji myGrenton:

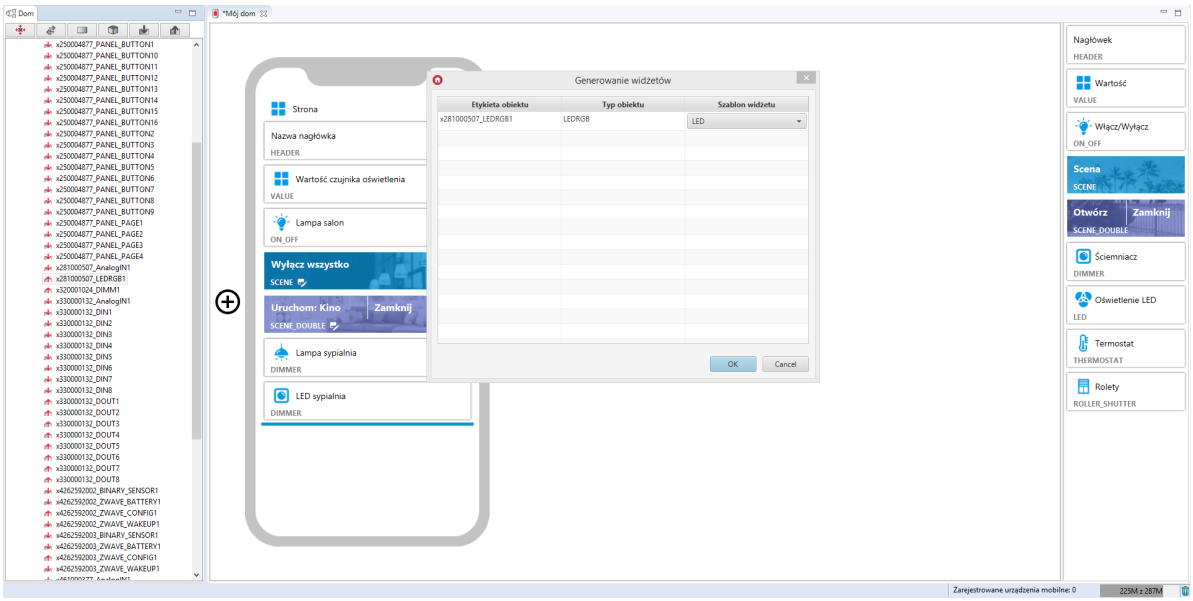


3.11. Oświetlenie LED (LED)

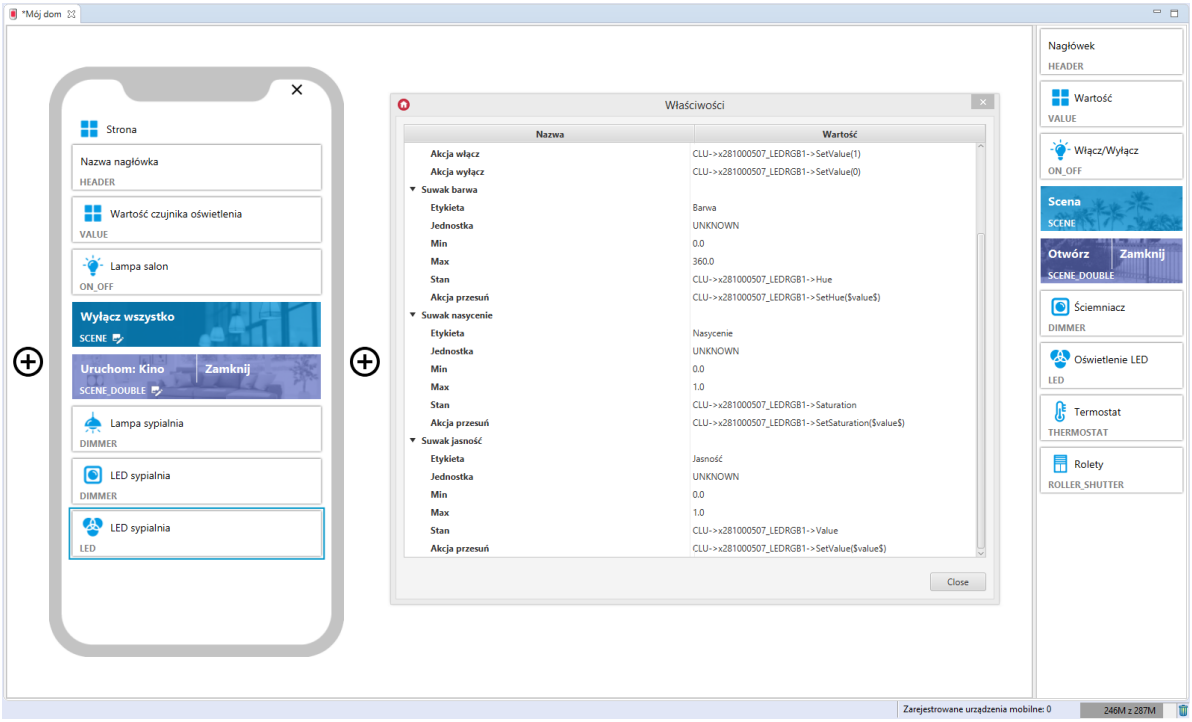
Widget dedykowany do oświetlenia LED. Posiada 1 przycisk bistabilny, działający na zasadzie załącz/wyłącz oraz 3 suwaki sterujące barwą, nasyceniem i jasnością.



Dla oświetlenia LED zdefiniowane są gotowe szablony dla widgetu LED. Aby dodać widget LED z gotowym szablonem należy przeciągnąć obiekt LEDRGB z listy obiektów na stronę interfejsu:



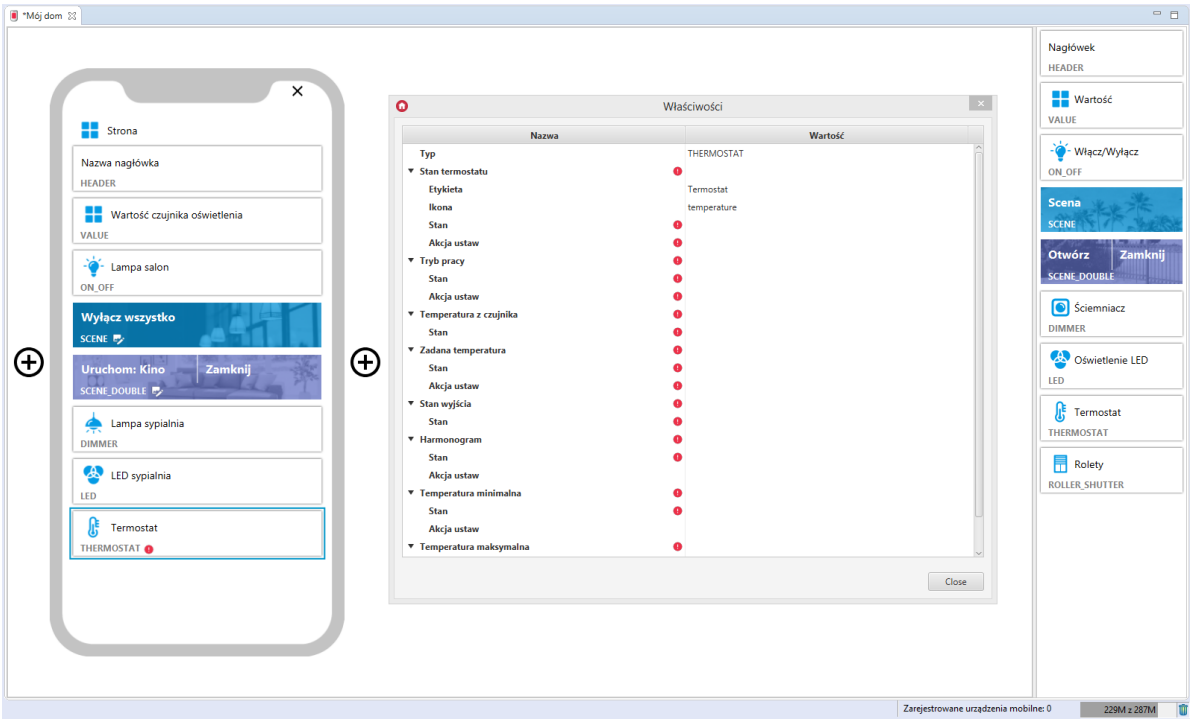
Skonfigurowany widget LED:



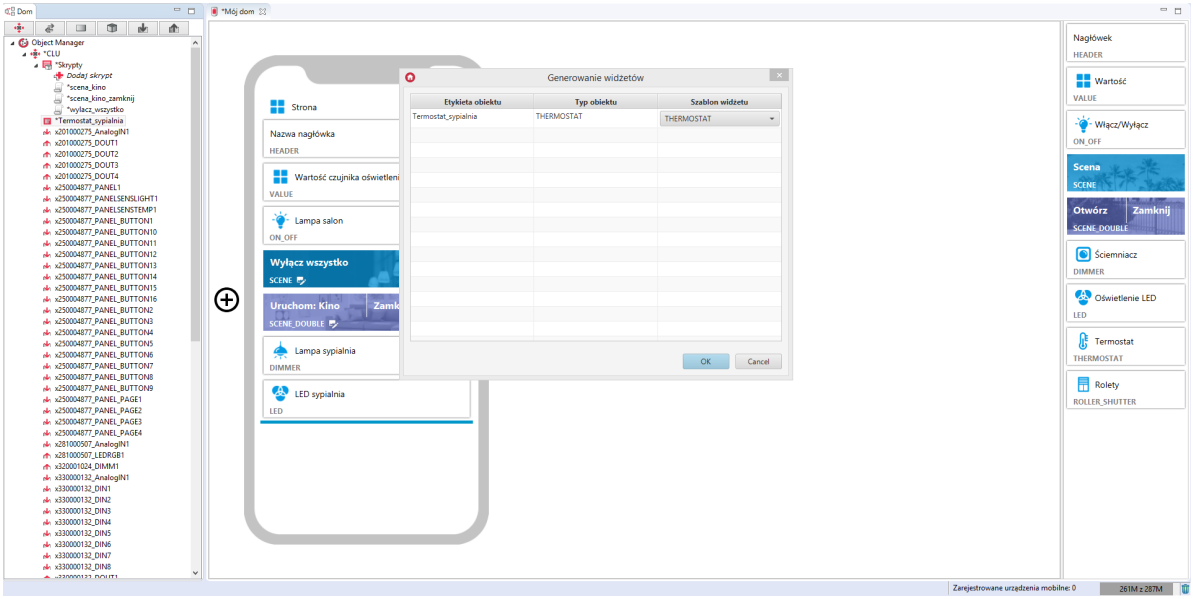
3.12. Termostat (THERMOSTAT)

> **Uwaga!** Widget obsługiwany jest dla termostatów utworzonych w **CLU 2.0!**

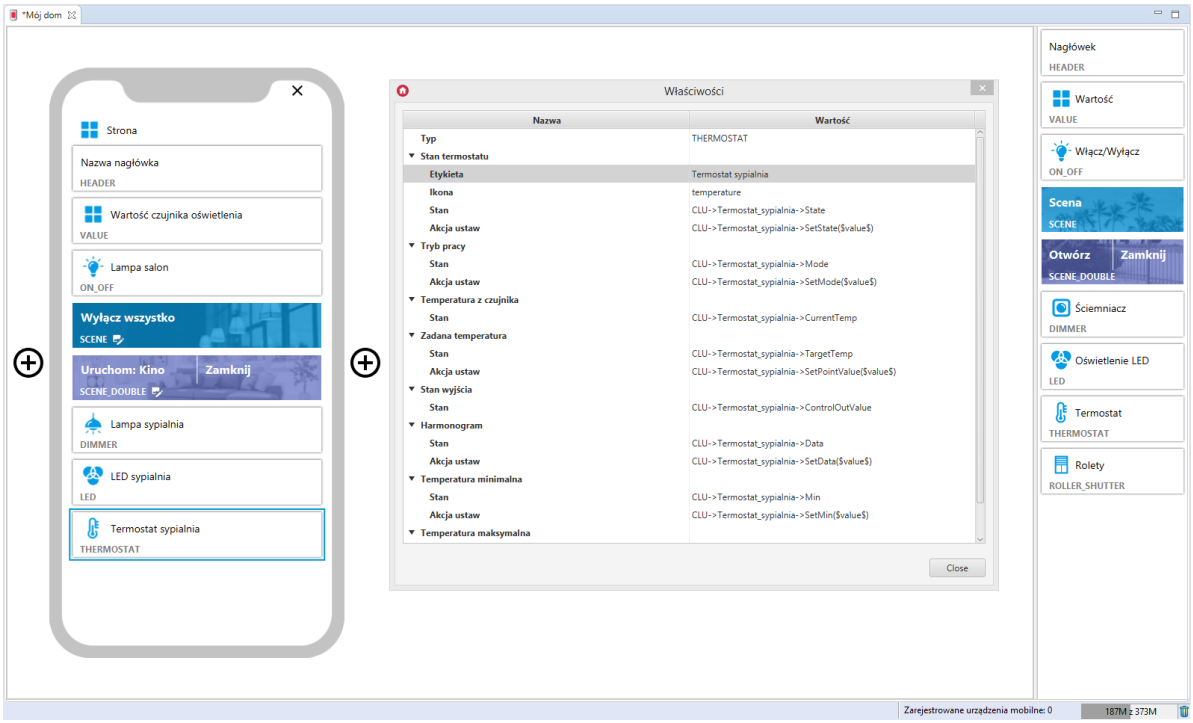
Widget dedykowany dla obiektów wirtualnych typu termostat. W przypadku, gdy przeciągamy z listy obiektów zdefiniowany już termostat do interfejsu, utworzony widget jest uzupełniony w oparciu o podane cechy wejścia i wyjścia termostatu.



Dla termostatów zdefiniowane są gotowe szablony dla widgetu THERMOSTAT. Aby dodać widget THERMOSTAT z gotowym szablonem należy przeciągnąć obiekt wirtualny termostat z listy obiektów na stronę interfejsu:



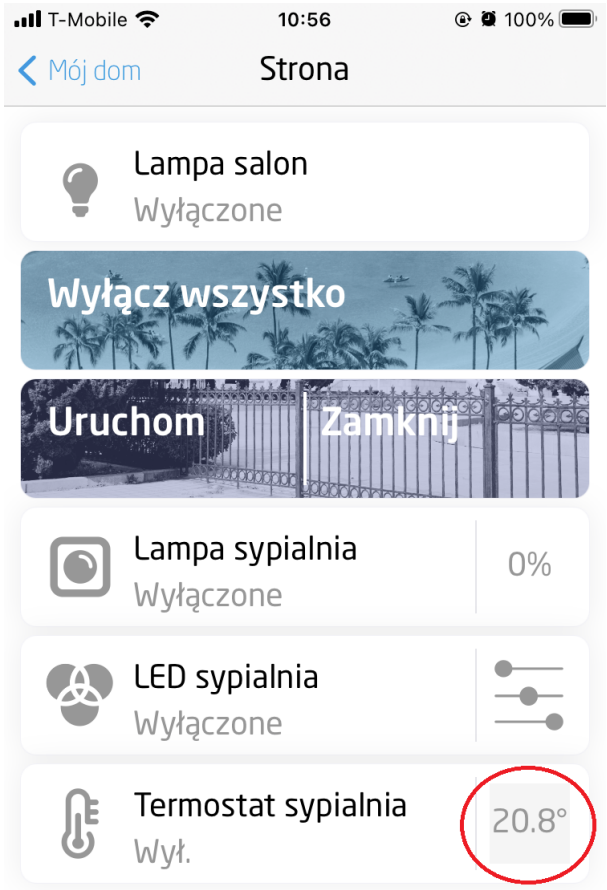
Skonfigurowany widget THERMOSTAT:



A. Konfiguracja harmonogramu w aplikacji

> **Uwaga!** Nowa konfiguracja harmonogramu dostępna jest dla aplikacji myGrenton w wersji 1.2.3 lub wyższej (Android) i wersji 1.6.0 lub wyższej (iOS).

W aplikacji myGrenton istnieje możliwość edycji harmonogramu termostatu. Aby to zrobić, należy kliknąć w pole temperatury widgetu:

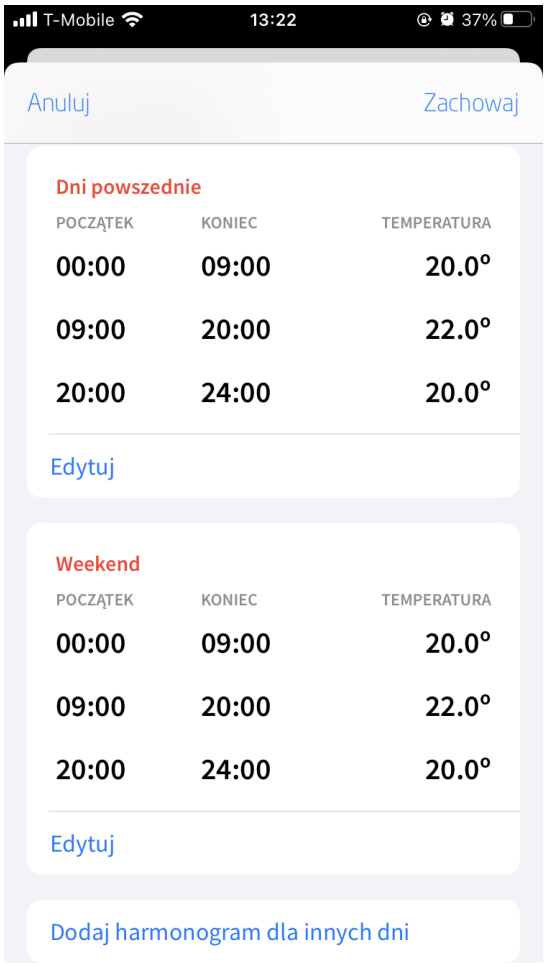


Następnie wybrać tryb harmonogramu i opcję `EDYTUJ HARMONOGRAM`:

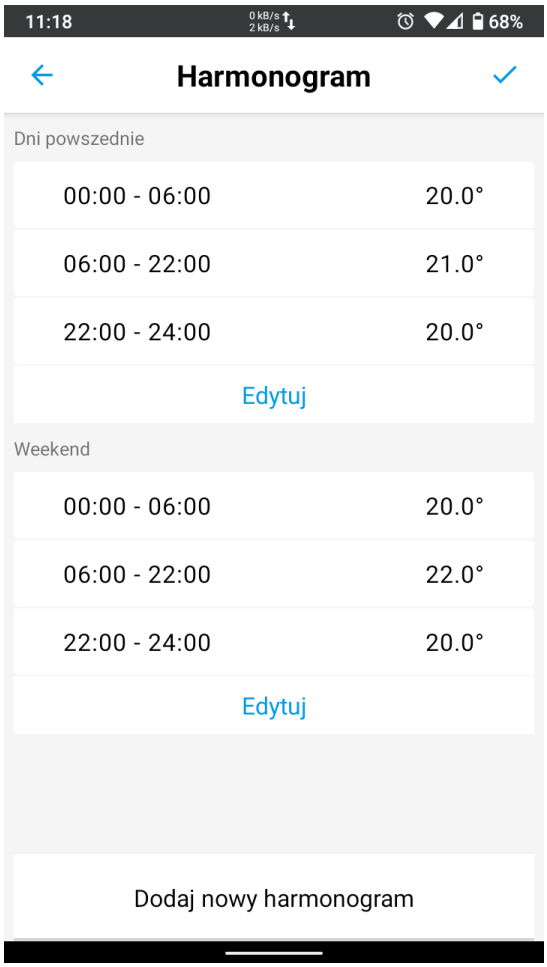


W oknie wyświetli się harmonogram pobrany z CLU. Istnieje możliwość edycji tego harmonogramu lub dodania nowych harmonogramów dla poszczególnych dni tygodnia:

iOS

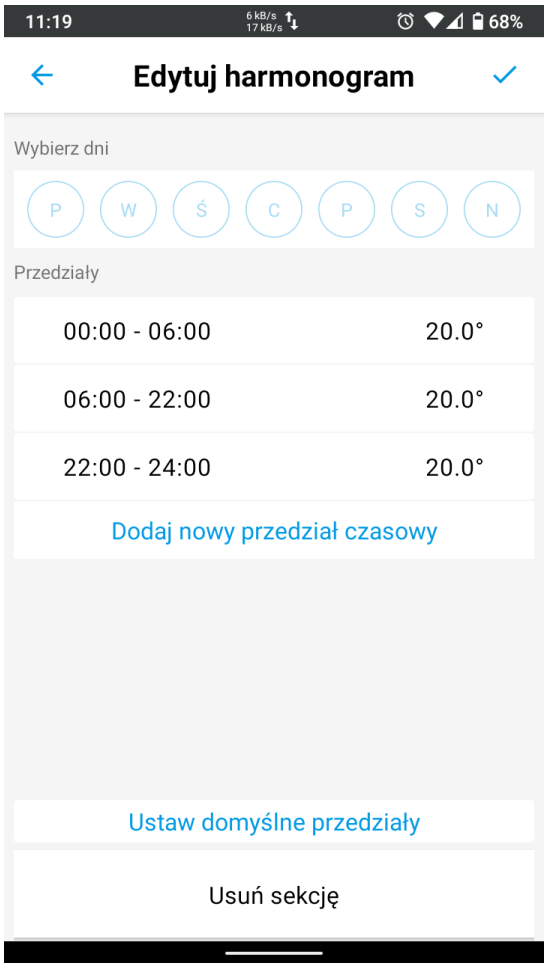


Android



Dodawanie nowego harmonogramu

Po wybraniu `Dodaj harmonogram dla innych dni` (iOS), bądź `Dodaj nowy harmonogram` (Android), otworzy się okno dodawania harmonogramu.

iOS	Android
	

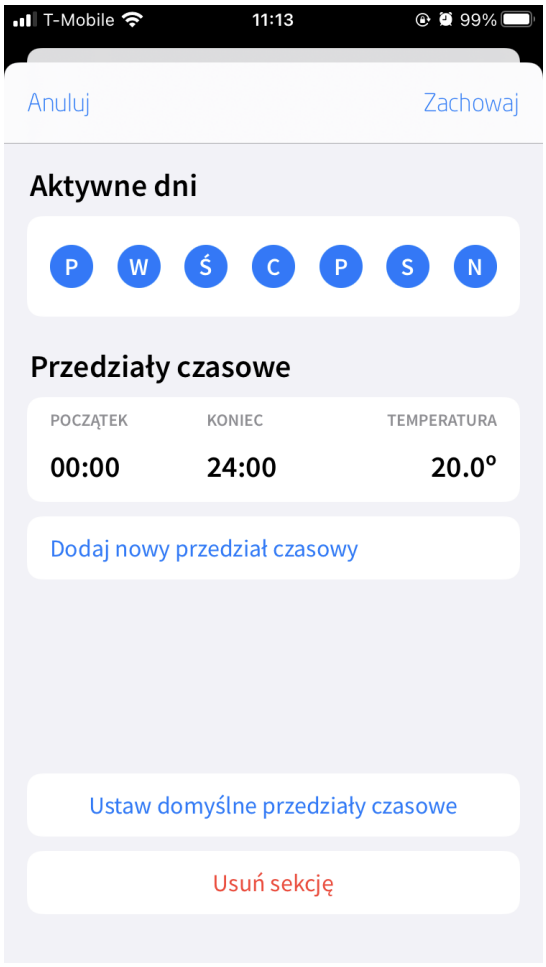
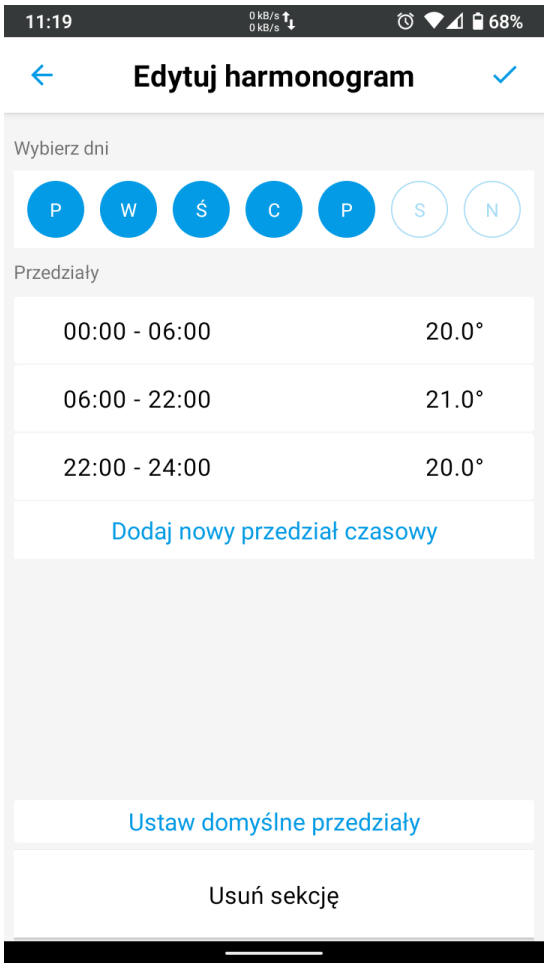
Należy:

- Zaznaczyć wybrane dni tygodnia (co najmniej jeden dzień musi być wybrany),
- Ustawić temperaturę dla określonych przedziałów czasowych (po otwarciu okna wyświetlone są domyślne przedziały czasowe),
- Akceptować zmiany klikając `Zachowaj`.

Aplikacja dla niezaznaczonych dni tygodnia automatycznie utworzy nowy harmonogram lub doda je do już istniejącego, aby prawidłowo uzupełnić wartości dla całego tygodnia.

Usuwanie / edytowanie sekcji harmonogramu

Po wybraniu `Edytuj` dla wybranej sekcji harmonogramu otworzy się okno edycji, w którym można edytować wybrane dni tygodnia, przedziały czasowe bądź usunąć daną sekcję.

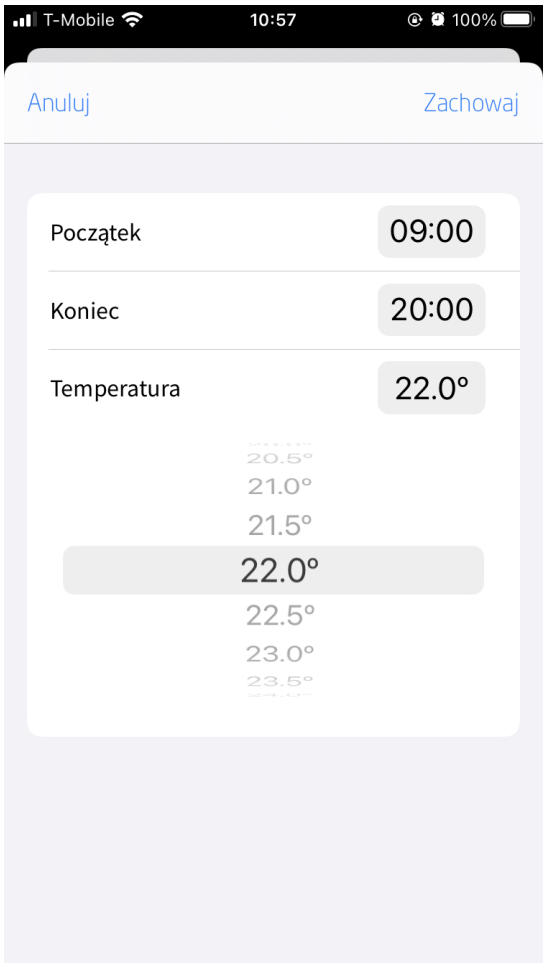
iOS	Android
	

Aby usunąć sekcję harmonogramu, należy kliknąć `Usuń sekcję`. Po usunięciu harmonogramu, dla brakujących dni tygodnia aplikacja automatycznie utworzy nowy harmonogram lub doda je do już istniejącego, aby prawidłowo uzupełnić wartości dla całego tygodnia.

Dzięki opcji `Ustaw domyślne przedziały czasowe`, można zastąpić aktualne przedziały czasowe domyślnymi.

Dodawanie nowych przedziałów czasowych

Po wybraniu `Dodaj nowy przedział czasowy`, otworzy się okno dodawania przedziału.

iOS	Android
	

Należy:

- Podać godzinę początku przedziału,
- Podać godzinę końca przedziału,
- Ustawić pożądaną temperaturę,
- Akceptować zmiany klikając `Zachowaj`.

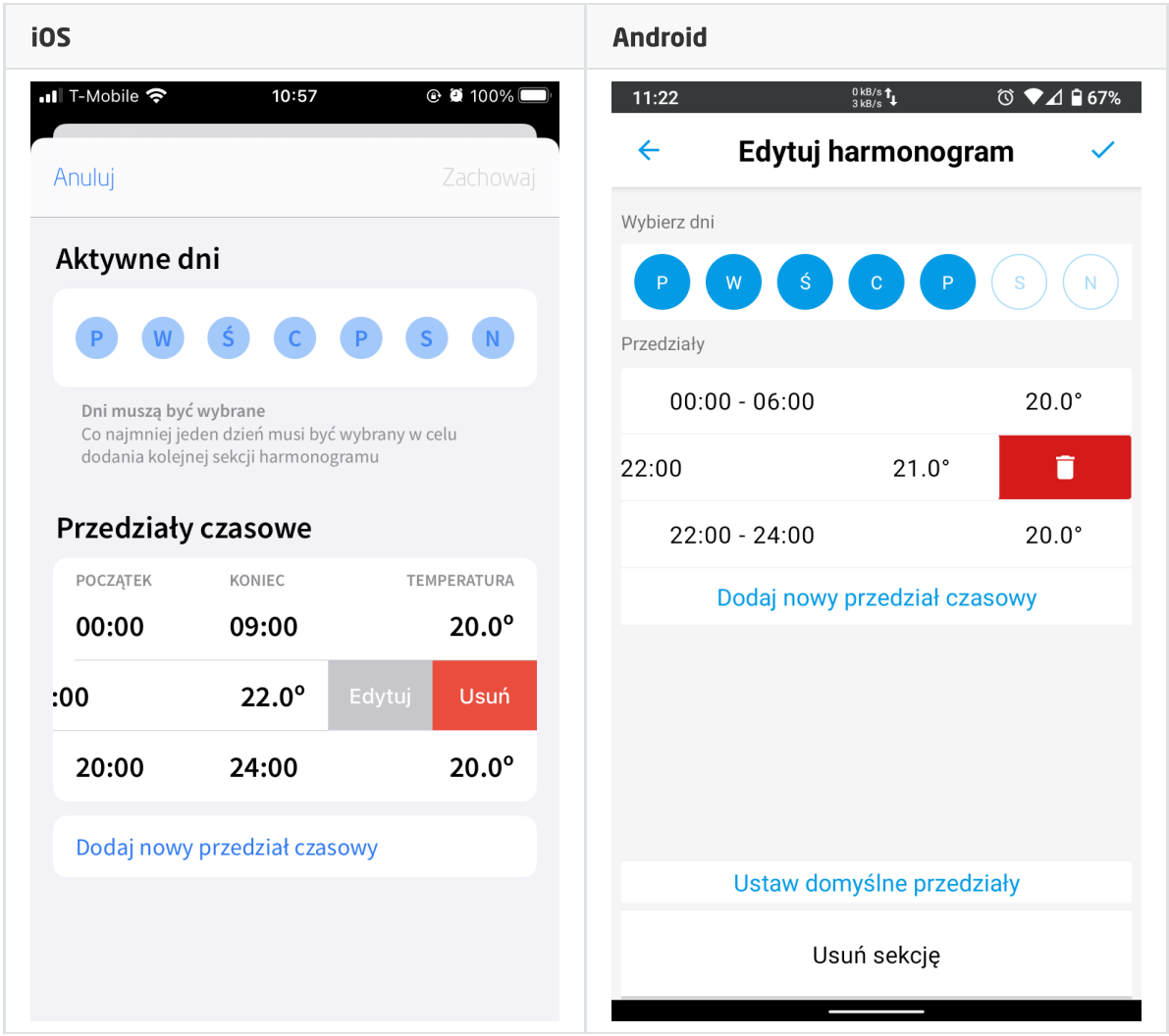
Aplikacja automatycznie doda przedziały dla nieuwzględnionych godzin, aby prawidłowo uzupełnić wartości dla całej doby.

> **Uwaga!** Opcja `Dodaj nowy przedział czasowy` umożliwia dodanie maksymalnie 6 przedziałów czasowych.

Usuwanie / edytowanie przedziału czasowego

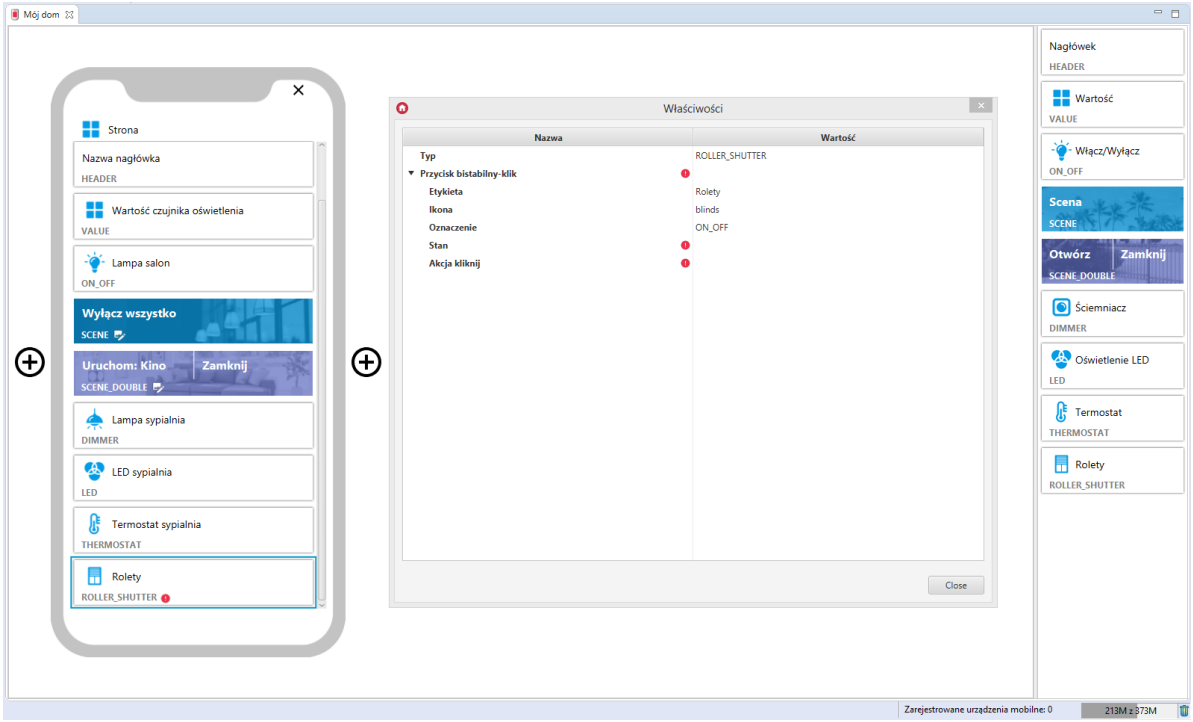
Aby przejść do edycji istniejącego przedziału należy kliknąć na przedział czasowy (iOS / Android) lub wykonać gest przesunięcia w lewo na przedziale czasowym, a następnie kliknąć na opcję `Edytuj` (opcja dostępna tylko dla iOS).

Aby usunąć przedział czasowy należy wykonać gest przesunięcia w lewo na przedziale czasowym, a następnie kliknąć opcję `Usuń`.

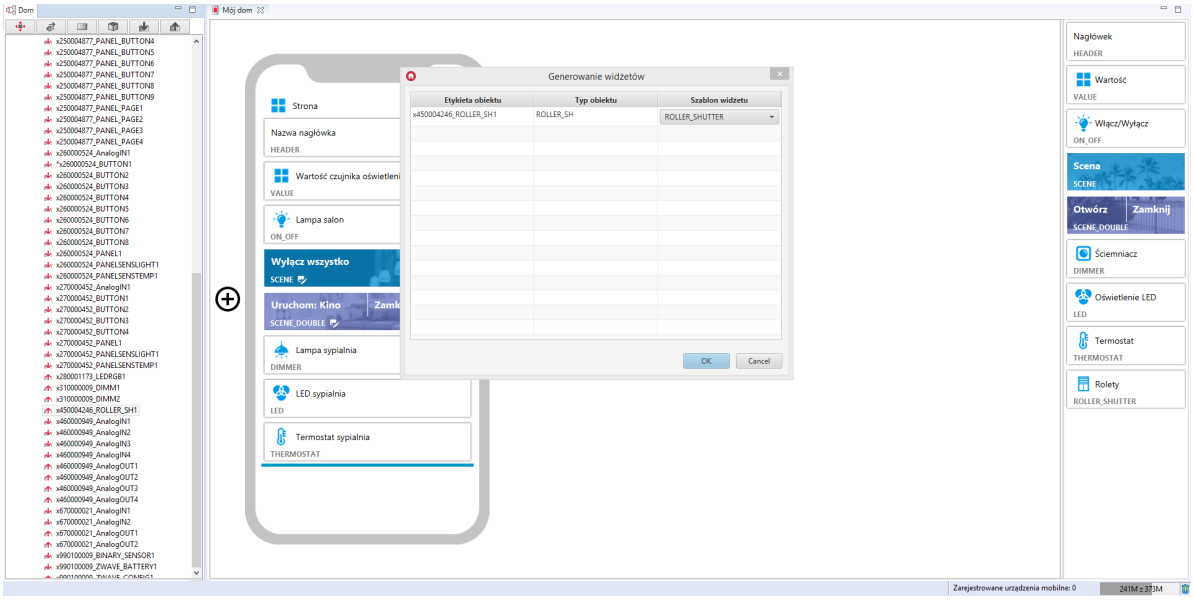


3.13. Rolety (ROLLER_SHUTTER)

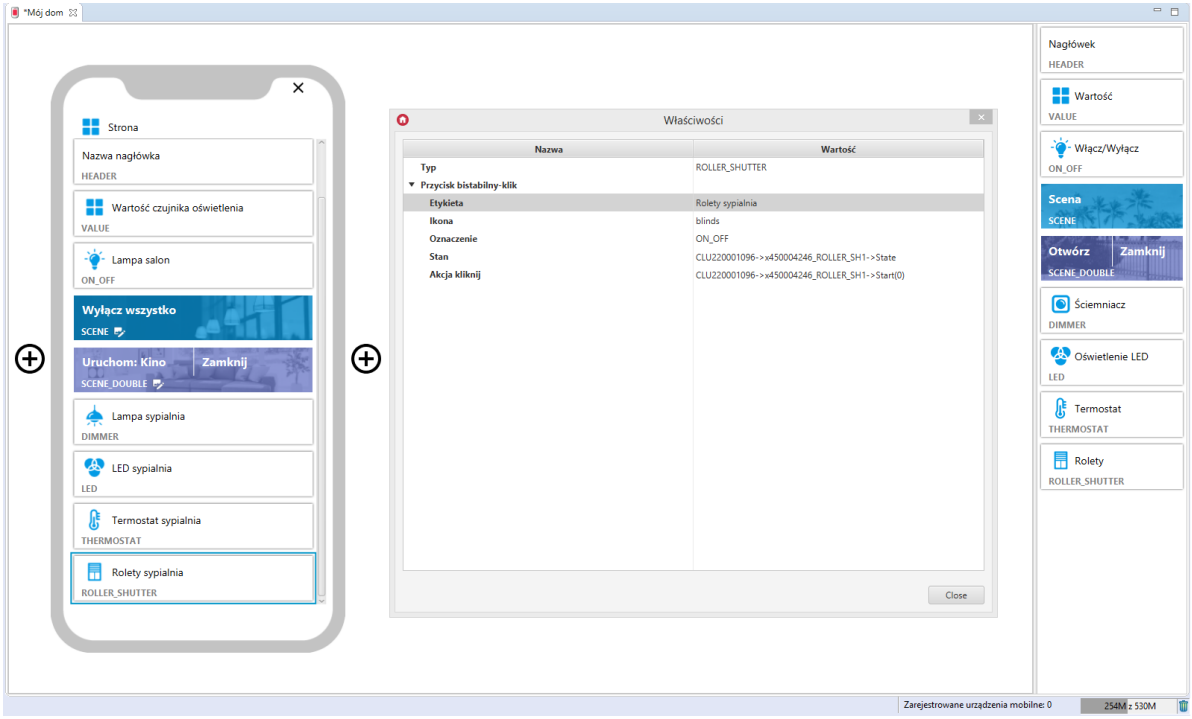
Widget służy do sterowania roletami. Aplikacja wyświetla obecny stan rolet (ZATRZYMANE, ZAMYKANIE, OTWIERANIE).



Dla sterowników rolet zdefiniowane są gotowe szablony dla widgetu ROLLER_SHUTTER. Aby dodać widget ROLLER_SHUTTER z gotowym szablonem należy przeciągnąć obiekt ROLLER_SHUTTER z listy obiektów na telefon:



Skonfigurowany widget ROLLER_SHUTTER:



3.14. Rolety v2 (ROLLER_SHUTTER_V2)

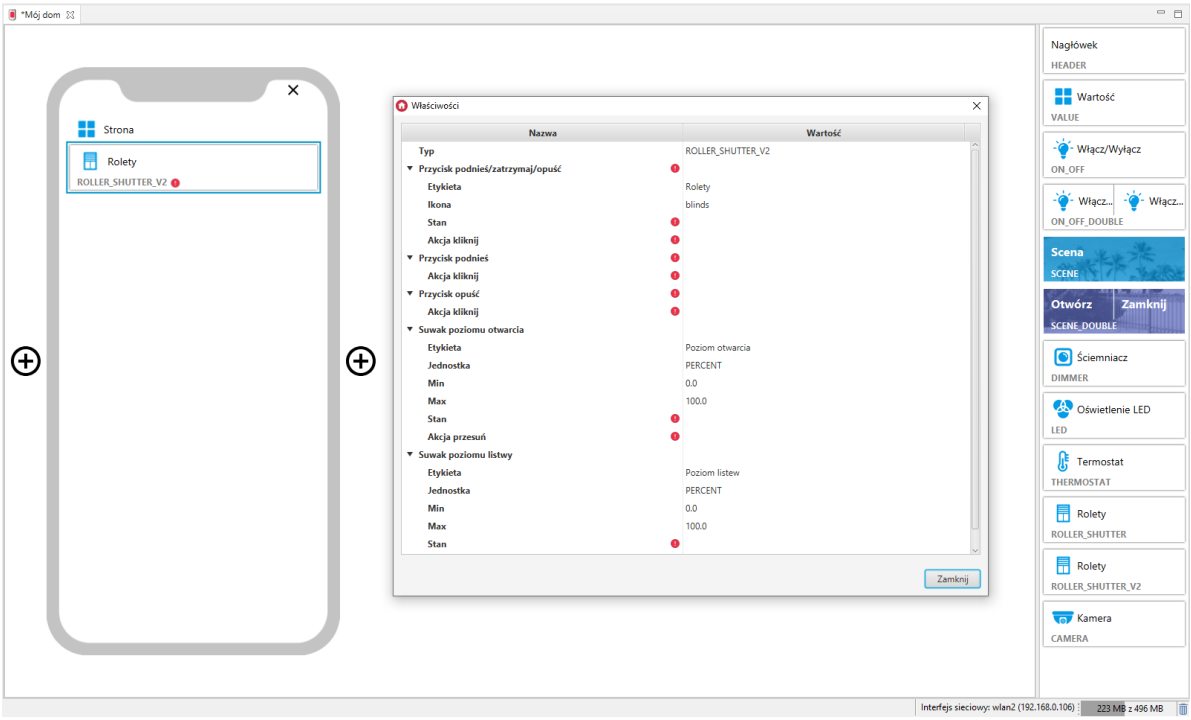
Uwaga! Widget ROLLER_SHUTTER_V2 dostępny jest dla Object Manager w wersji 1.4.0 lub wyższej, CLU w wersji 5.7.1 lub wyższej, modułu Roller Shutter x1 DIN / Roller Shutter x3 DIN / Roller Shutter FM w wersji 2.1.1 lub wyższej oraz dla aplikacji myGrenton w wersji 1.2.3 (Android) / 1.6.0 (iOS) lub wyższej.

Uwaga! Od wersji Object Manager 1.6.0 widget ROLLER_SHUTTER_V2 oraz możliwość wykorzystania go jako gotowego szablonu będą niedostępne. Zostaje on zastąpiony widgetem ROLLER_SHUTTER_V3.

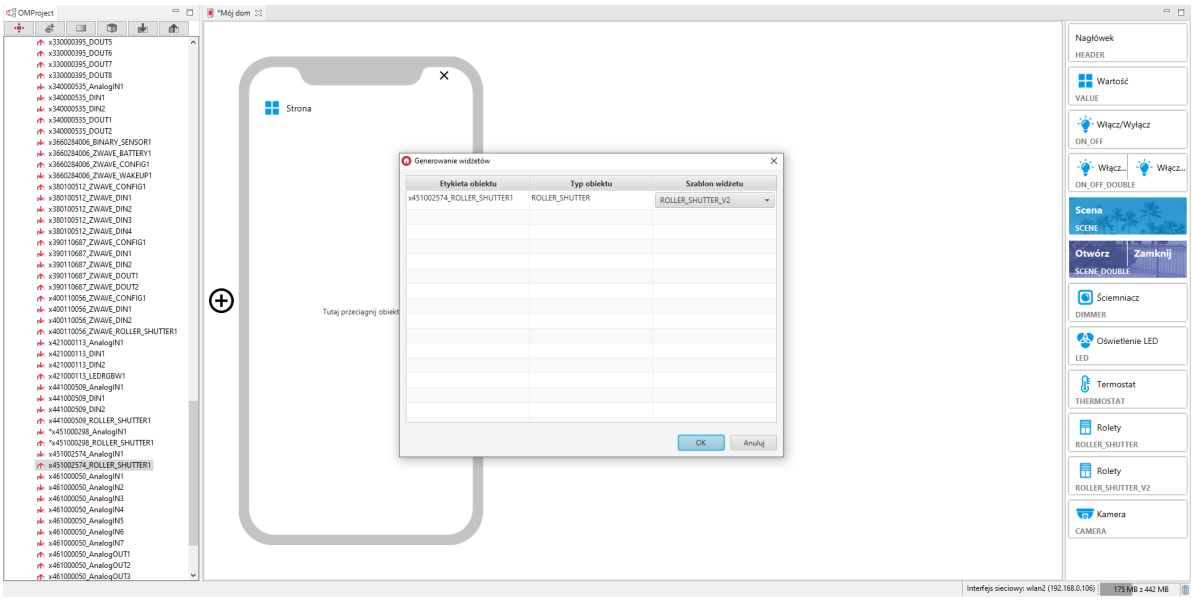
> Widgety ROLLER_SHUTTER_V2 znajdujące się w projektach stworzonych na poprzednich wersjach Object Manager będą nadal poprawnie obsługiwane i wyświetlane zarówno w Object Mangerze jak i w aplikacji myGrenton.

Widget dedykowany do zaawansowanego sterowania roletami oraz lamelami. Widget ROLLER SHUTTER_V2 zawiera:

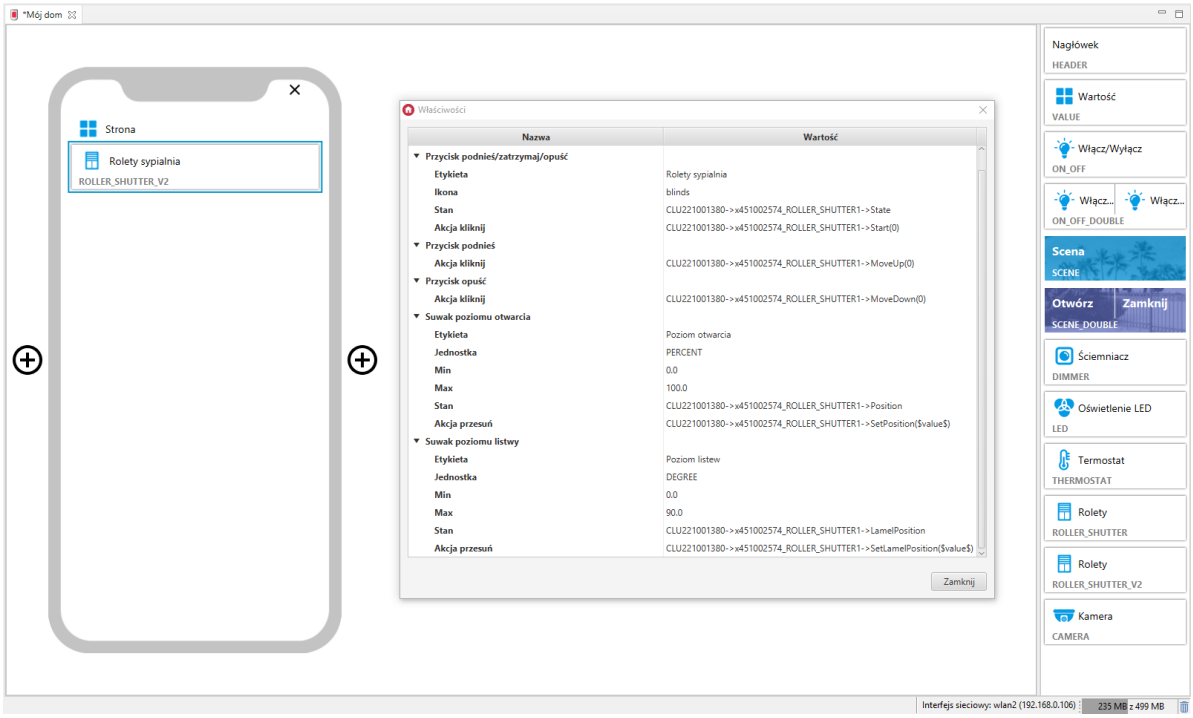
- Przyciski umożliwiające akcje OTWIERANIA / ZAMYKANIA / ZATRZYMANIA rolety,
- Procentowy poziom otwarcia rolety,
- Suwaki sterujące i wyświetlające poziom otwarcia rolety i lameli.



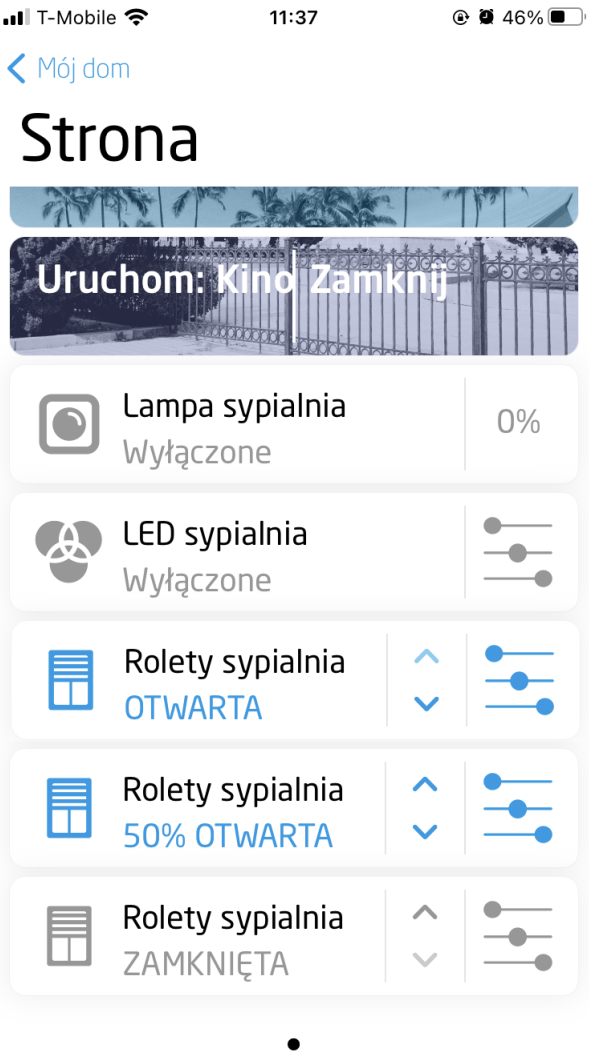
Dla sterowników rolet zdefiniowane są gotowe szablony dla widgetu ROLLER_SHUTTER_V2. Aby dodać widget ROLLER_SHUTTER_V2 z gotowym szablonem należy przeciągnąć obiekt ROLLER_SHUTTER z listy obiektów na telefon:

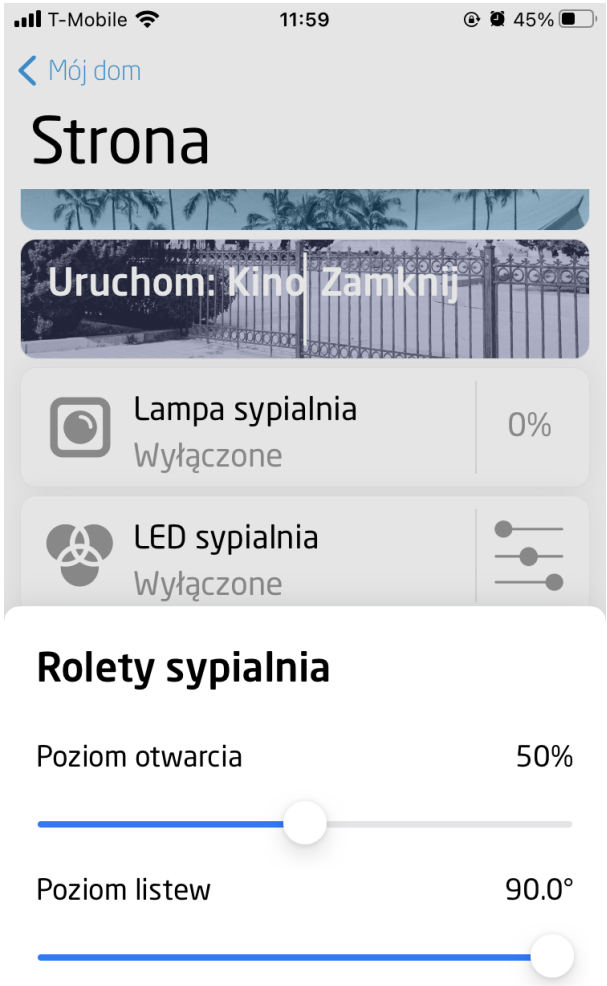


Skonfigurowany widget ROLLER_SHUTTER_V2:



Wygląd widgetu w aplikacji myGrenton:



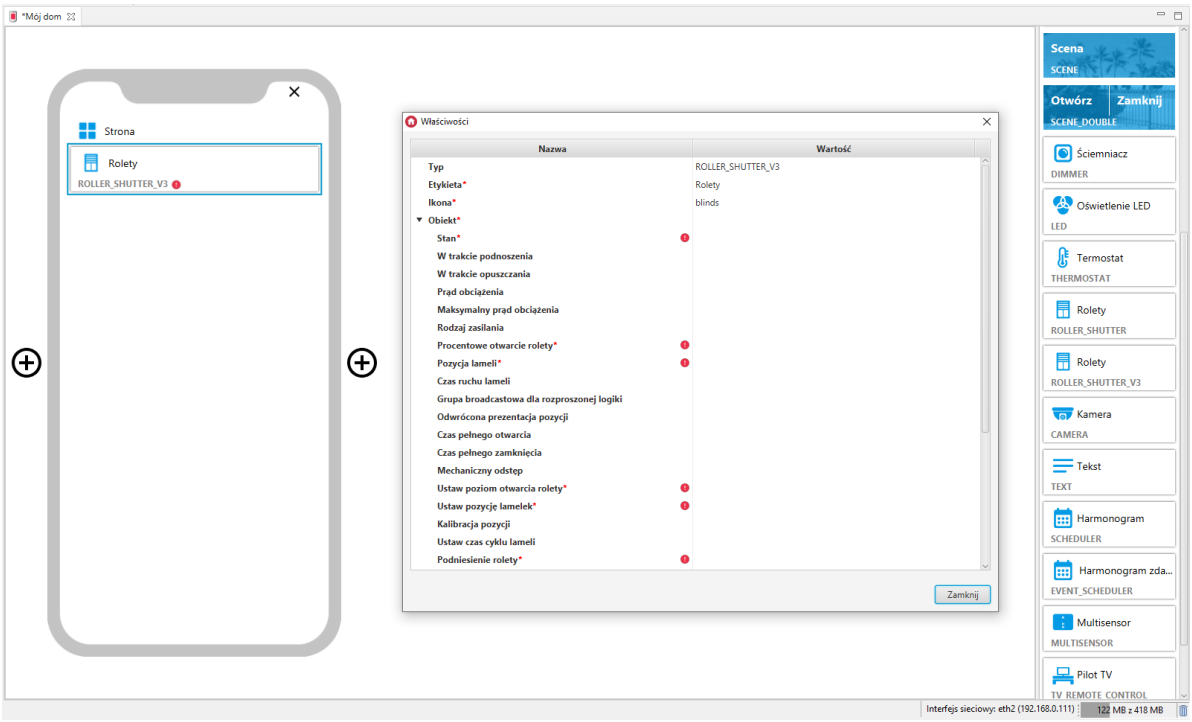


3.15. Rolety v3 (ROLLER_SHUTTER_V3)

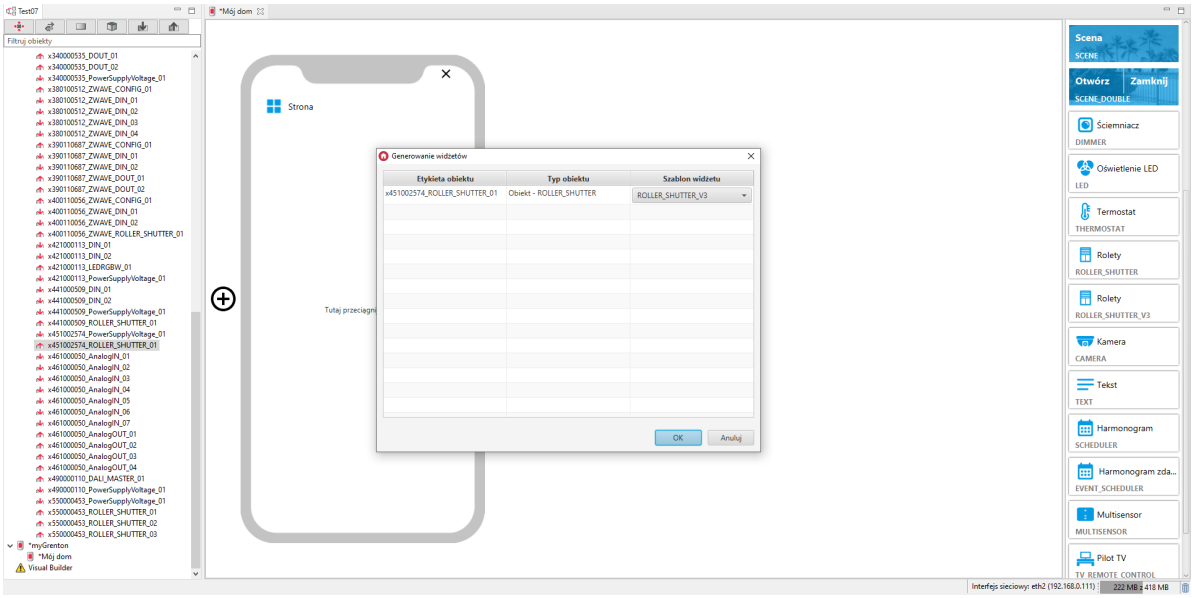
> **Uwaga!** Widget ROLLER_SHUTTER_V3 dostępny jest dla Object Manager w wersji 1.6.0 lub wyższej, CLU w wersji 5.7.1 lub wyższej, modułu Roller Shutter x1 DIN / Roller Shutter x3 DIN / Roller Shutter FM w wersji 2.1.1 lub wyższej oraz dla aplikacji myGrenton w wersji 1.4.0 (Android) / 1.8.0 (iOS) lub wyższej.

Widget dedykowany do zaawansowanego sterowania roletami oraz lamelami. Widget ROLLER SHUTTER_V3 zawiera:

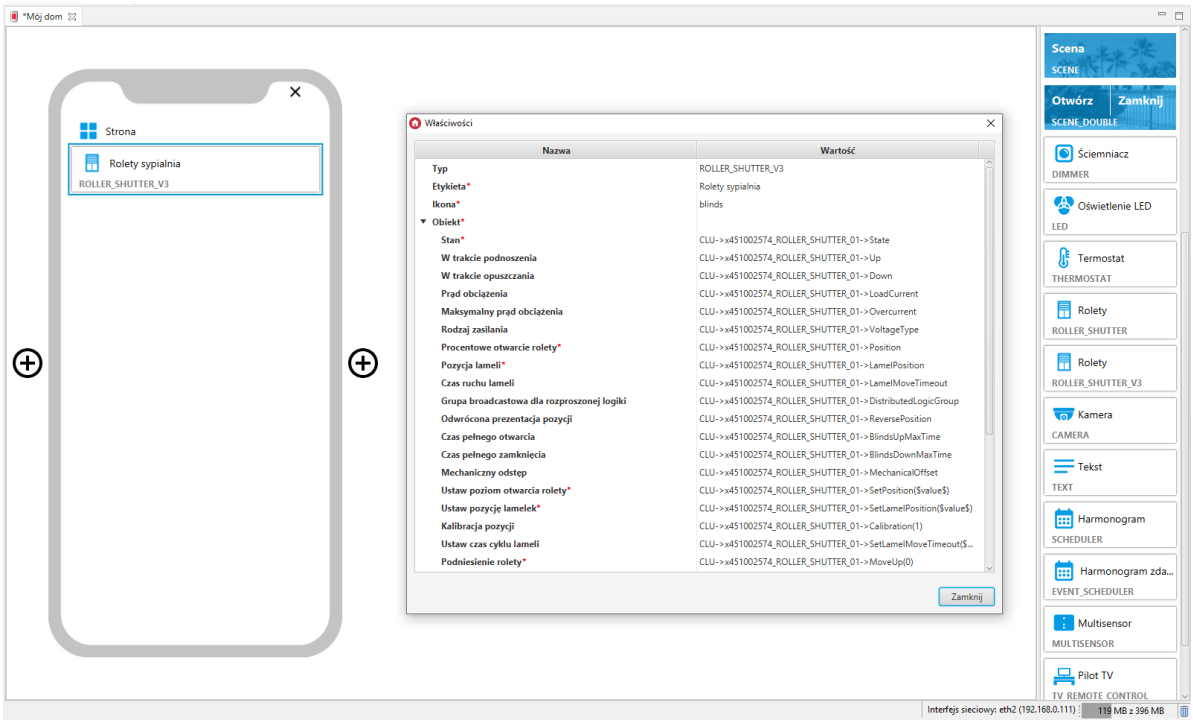
- Przyciski umożliwiające akcje OTWIERANIA / ZAMYKANIA / ZATRZYMANIA rolety,
- Procentowy poziom otwarcia rolety,
- Suwaki sterujące i wyświetlające poziom otwarcia rolety i lameli,
- Wyświetlanie stanów `Kalibracja` / `Zablokowana`.



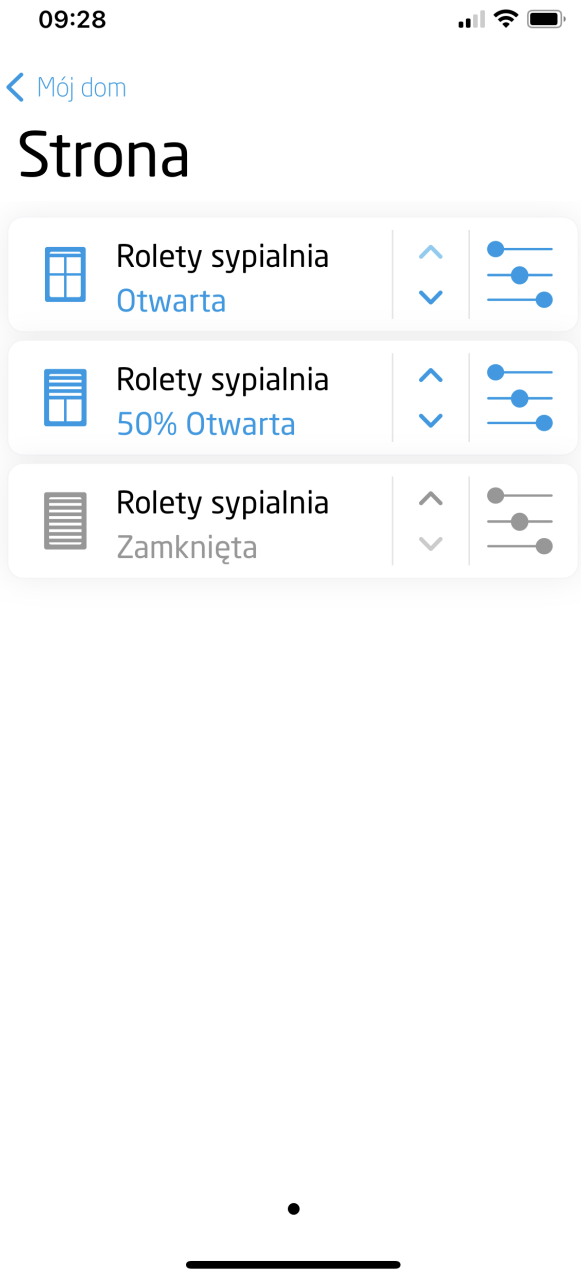
Dla sterowników rolet zdefiniowane są gotowe szablony dla widgetu ROLLER_SHUTTER_V3. Aby dodać widget ROLLER_SHUTTER_V3 z gotowym szablonem należy przeciągnąć obiekt ROLLER_SHUTTER z listy obiektów na telefon:

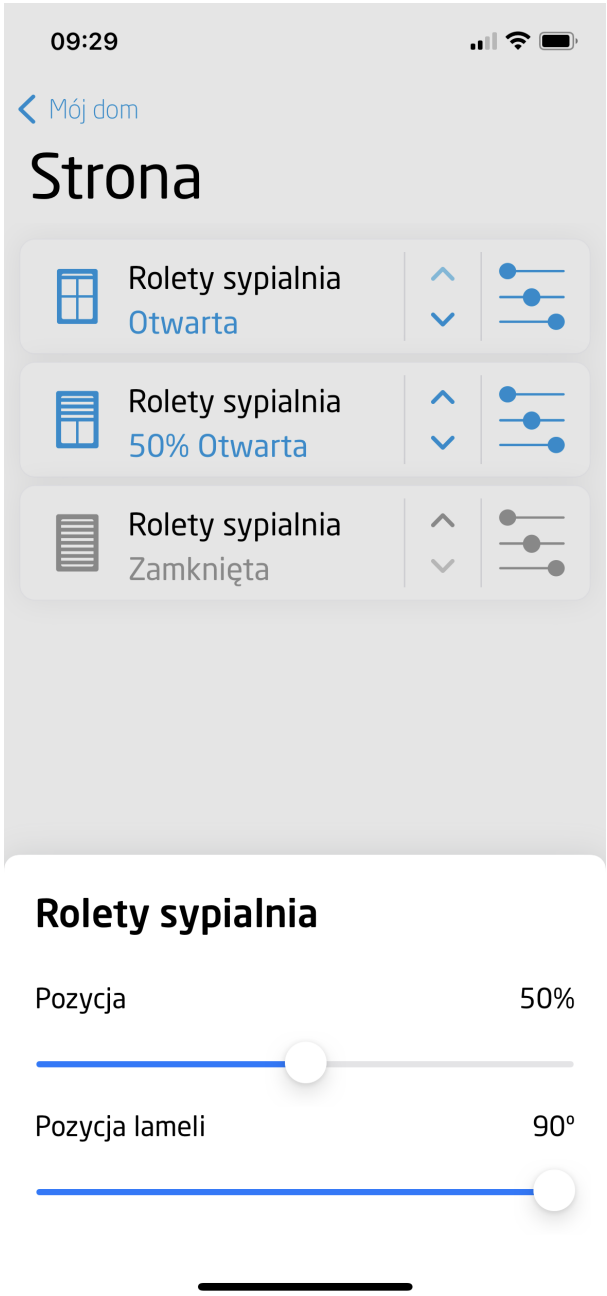


Skonfigurowany widget ROLLER_SHUTTER_V3:



Wygląd widgetu w aplikacji myGrenton:

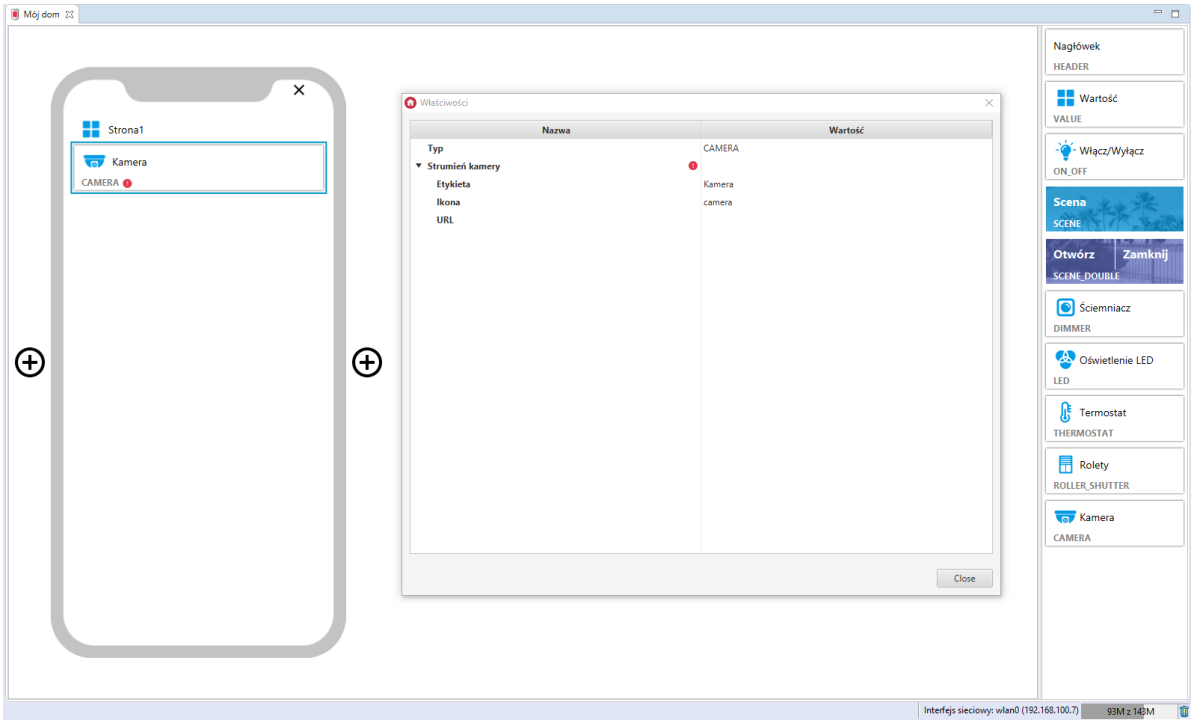

09:28
Mój dom
Strona
Rolety sypialnia
Otwarta
Rolety sypialnia
50% Otwarta
Rolety sypialnia
Zamknięta



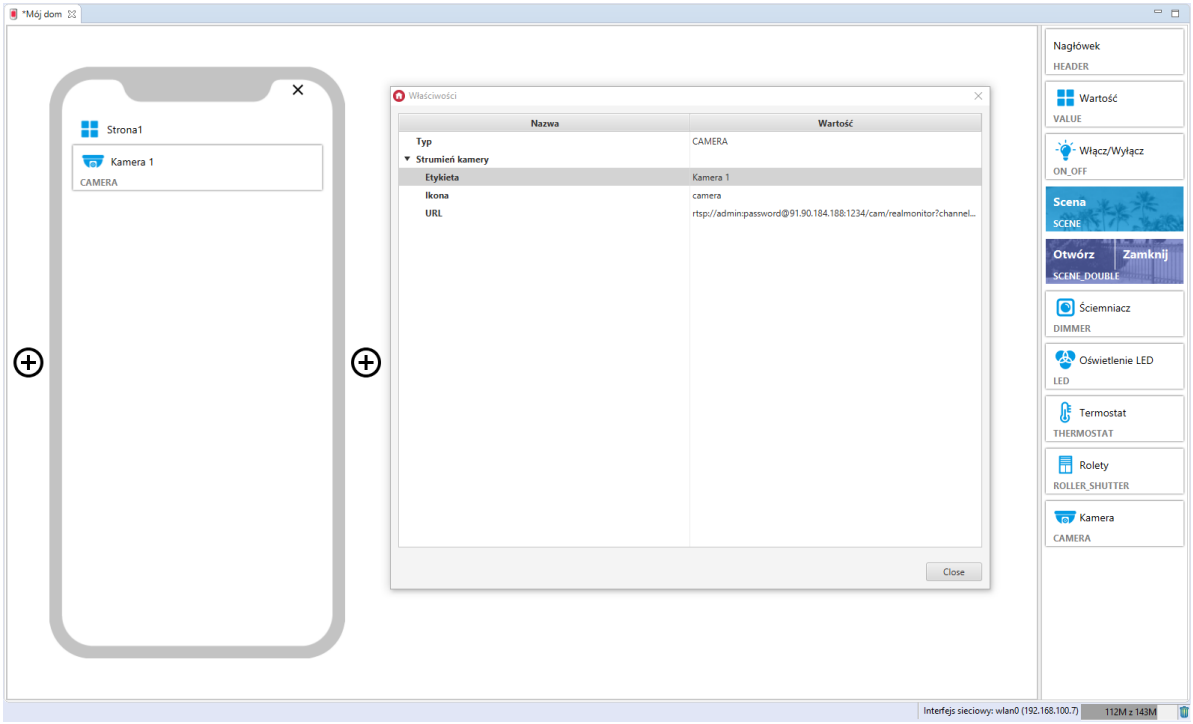
3.16. Kamera (CAMERA)

> **Uwaga!** Widget CAMERA dostępny jest dla Object Manager w wersji 1.3.5-204201 lub wyższej, oraz dla aplikacji myGrenton w wersji 1.2.2 lub wyższej (Android) i wersji 1.5.1 lub wyższej (iOS).

Widget służy do wyświetlania obrazu z kamery IP. Przeciągając widget z zakładki z prawej strony ekranu należy go uzupełnić o wartość w polu `URL` (Adres strumienia RTSP dla kamery IP).



Skonfigurowany widget CAMERA:



A. Konfiguracja kamery

Warunkiem poprawnego działania widgetu jest użycie strumienia video po protokole RTSP oraz nadawanie w kodeku MJPG. Szczegółowe informacje odnośnie konfiguracji danej kamery są określone indywidualnie w instrukcji danego urządzenia.

> **Uwaga!** Zaleca się użycie kamer marki Dahua lub Hikvision. Przykład konfiguracji przedstawiono na podstawie ustawień dla modeli:
>
> - Dahua IP CAMERA Model: IPC-HFW2531S-S-0280B-S2
> - Hikvision NETWORK CAMERA Model: DS-2CD1043G0-I

Przykład, jak uzyskać strumień RTSP dla kamery Hikvision:

1. RTSP bez uwierzytelnienia

   `rtsp: // < adres IP urządzenia > : < port RTSP > / Streaming / channels / < numer kanału > < numer strumienia > / ?transportmode=unicast`

   Uwaga: `< numer strumienia >` reprezentuje strumień główny (01) lub strumień podrzędny (02)

   Przykład:

   `rtsp://173.200.91.70:10554/Streaming/channels/101/?transportmode=unicast` - pobierz główny strumień pierwszego kanału

   `rtsp://173.200.91.70:10554/Streaming/channels/102/?transportmode=unicast` - pobierz podstrumień

2. RTSP z uwierzytelnianiem

   `rtsp: // < nazwa użytkownika > : < hasło > @ < adres IP urządzenia > : < port RTSP > / Streaming / channels / < channel numer > < numer strumienia > / ? transportmode=unicast`

   Przykład:

   `rtsp://admin:password@173.200.91.70:10554/Streaming/channels/101/? transportmode=unicast` - pobierz główny strumień

   `rtsp://admin:password@173.200.91.70:10554/Streaming/channels/102/? transportmode=unicast` - pobierz podstrumień

Przykład, jak uzyskać strumień RTSP dla kamery Dahua:

1. RTSP bez uwierzytelnienia

   `rtsp: // < adres IP urządzenia > : < port RTSP > / cam / realmonitor ? channel = < numer kanału > & subtype = < numer strumienia >`

   Przykład:

   `rtsp://173.200.91.70:10554/cam/realmonitor?channel=1&subtype=1` - pobierz główny strumień

2. RTSP z uwierzytelnianiem

   `rtsp: // < nazwa użytkownika > : < hasło > @ < adres IP urządzenia > : < port RTSP > / cam / realmonitor ? channel = < numer kanału > & subtype = < numer strumienia >`

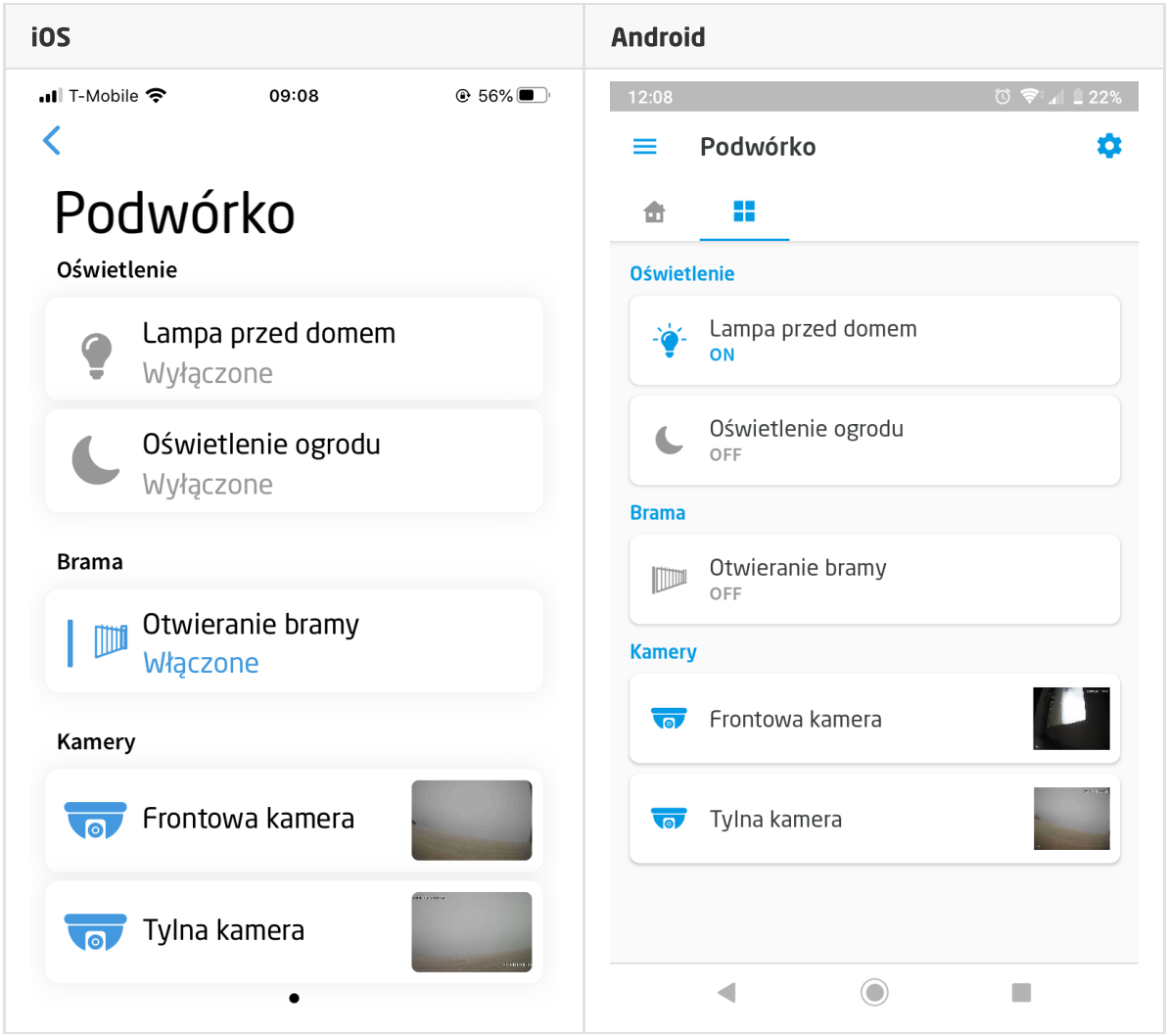
   Przykład:

   `rtsp://admin:password@173.200.91.70:10554/cam/realmonitor?channel=1&subtype=1` - pobierz główny strumień

B. Działanie widgetu w aplikacji

Obraz z kamery wyświetlany jest po naciśnięciu na widget w aplikacji. Aby zamknąć podgląd, należy nacisnąć krzyżyk w lewym górnym rogu ekranu, bądź użyć systemowego klawisza wstecz. Miniatura obrazu z kamery aktualizowana jest każdorazowo po wyświetleniu obrazu z kamery.

Wygląd widgetu kamery:



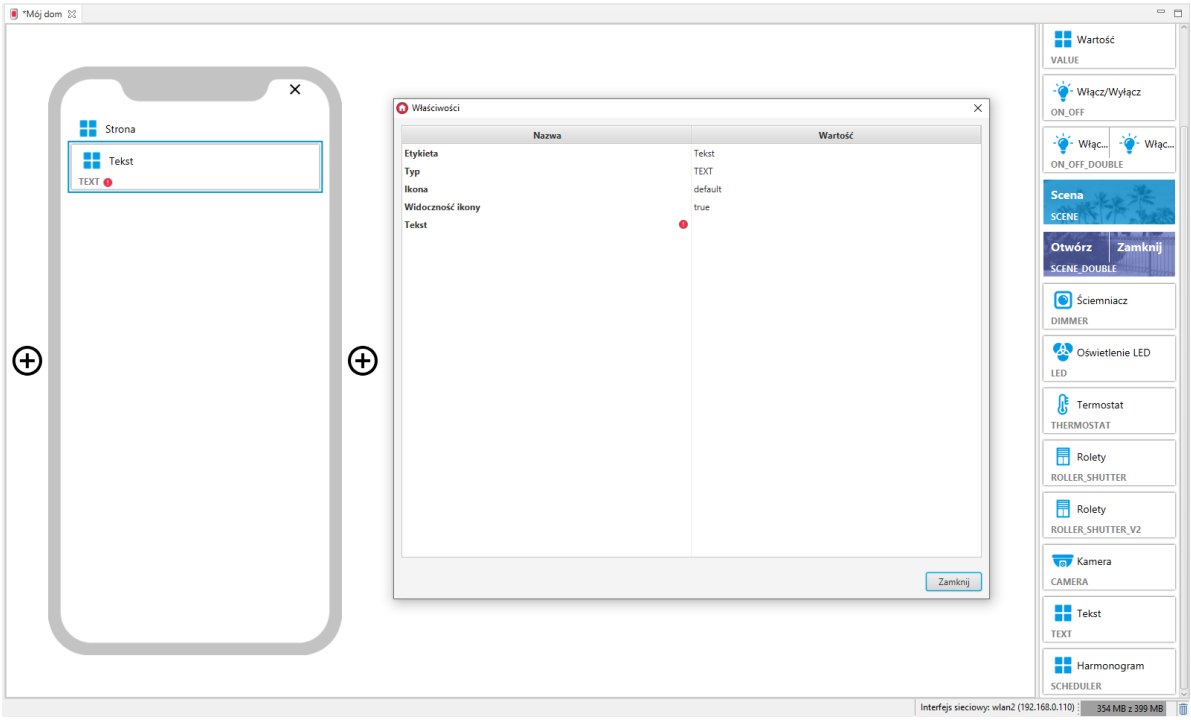
Widok podglądu obrazu z kamery:

iOS	Android
	

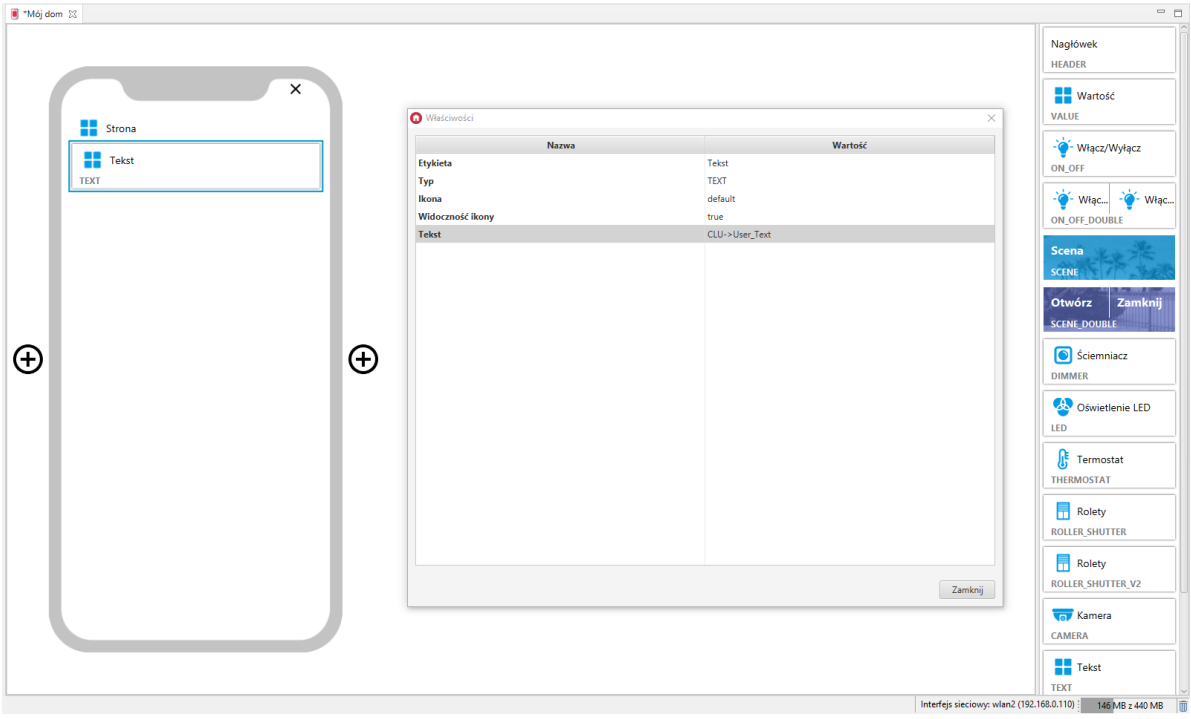
3.17. Tekst (TEXT)

> **Uwaga!** Widget TEXT dostępny jest dla Object Manager w wersji 1.5.0 lub wyższej, dla aplikacji myGrenton w wersji 1.3.0 (Android) / 1.7.0 (iOS) lub wyższej.

Widget dedykowany jest do wyświetlania wartości cechy użytkownika lub cechy wbudowanej danego obiektu. Widget dysponuje właściwością `Widoczność ikony`, która umożliwia wyświetlanie lub ukrycie ikony widgetu w aplikacji.



Skonfigurowany widget TEXT:



Widget wspiera podstawowe znaczniki języka HTML:

- `<br>` - Złamanie wiersza,
- `<b>` - Czcionka pogrubiona,
- `<i>` - Czcionka pochylona,
- `<em>` - Wyróżnienie w kolorze motywu interfejsu.

Możliwe jest również zagnieżdżanie znaczników.

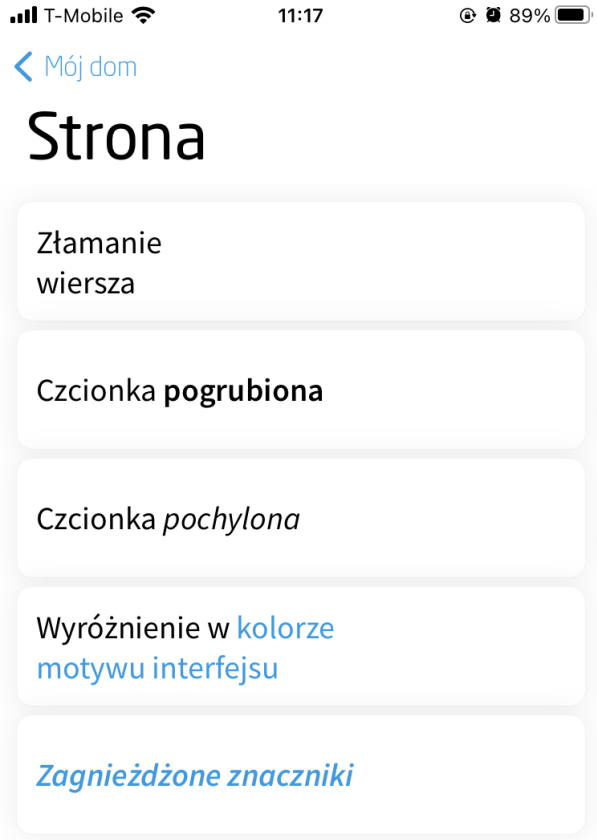
Przykłady użycia znaczników:

- Utworzenie cech użytkownika typu String, np:
  - `string_br` Wartość początkowa: `Złamanie <br>wiersza` ,
  - `string_b` Wartość początkowa: `Czcionka <b>pogrubiona</b>` ,
  - `string_i` Wartość początkowa: `Czcionka <i>pochylona</i>` ,

- `string_em` Wartość początkowa: `Wyróżnienie w <em>kolorze motywu interfejsu</em>`.
- `string_zagniezdzony` Wartość początkowa: `<b><i><em>Zagnieżdżone znaczniki</em></i></b>`.

- Utworzenie interfejsu myGrenton zawierającego skonfigurowane widgety TEXT, dla stworzonych cech użytkownika.
- Wysłanie interfejsu na urządzenie mobilne.

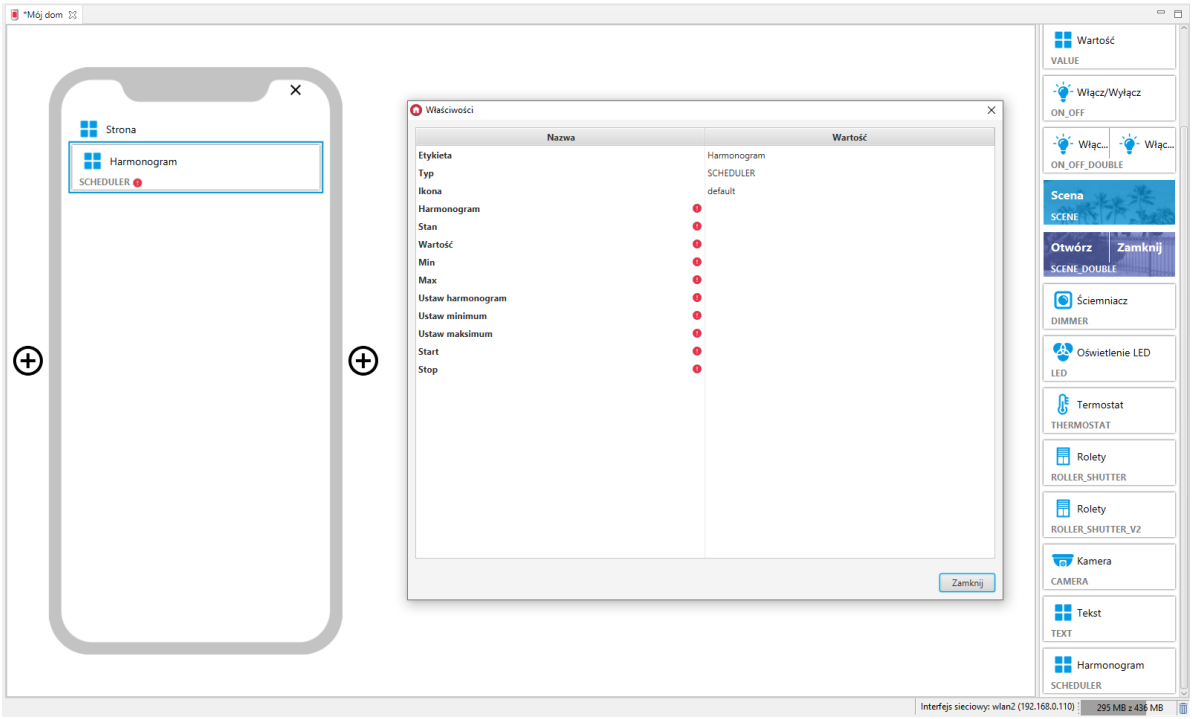
Wygląd w aplikacji:



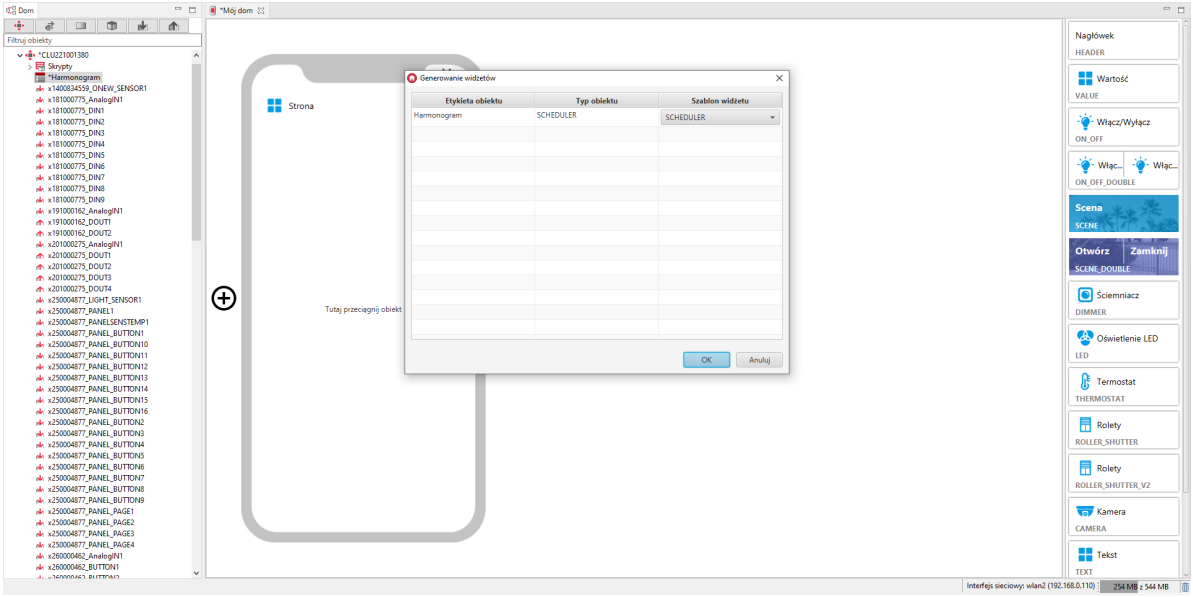
3.18. Harmonogram (SCHEDULER)

> **Uwaga!** Widget SCHEDULER dostępny jest dla Object Manager w wersji 1.5.0 lub wyższej, dla aplikacji myGrenton w wersji 1.3.0 (Android) / 1.7.0 (iOS) lub wyższej.

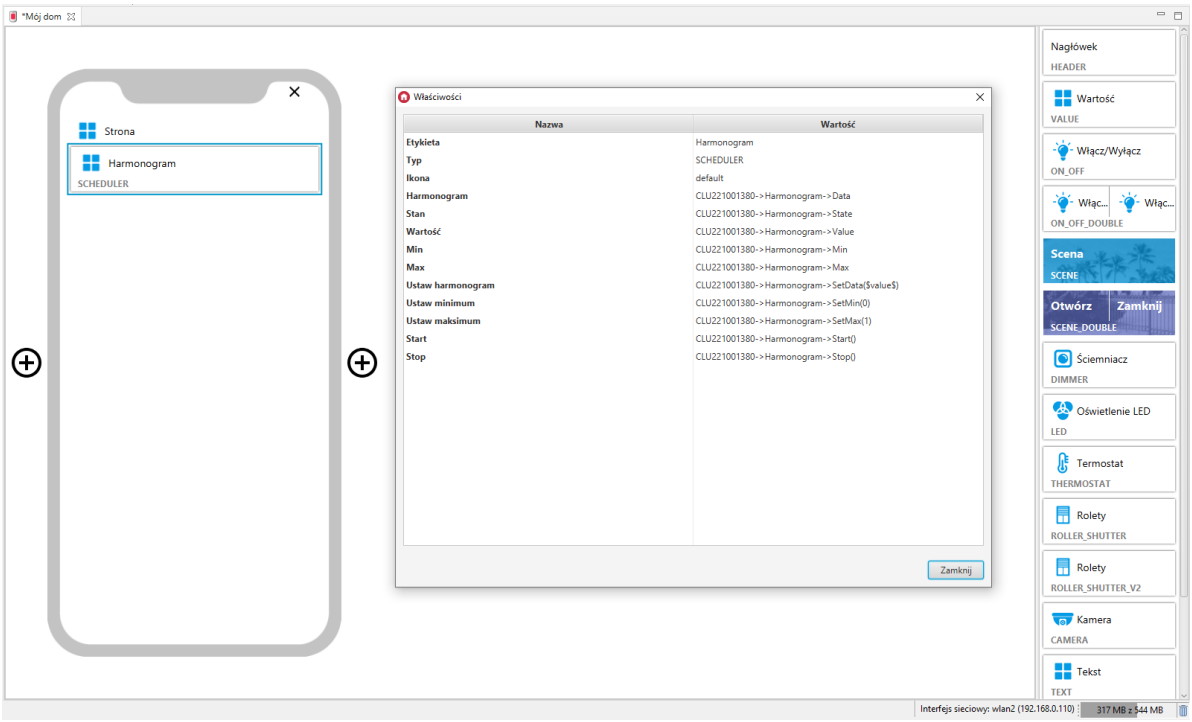
Widget dedykowany dla obiektów wirtualnych typu harmonogram (Scheduler).



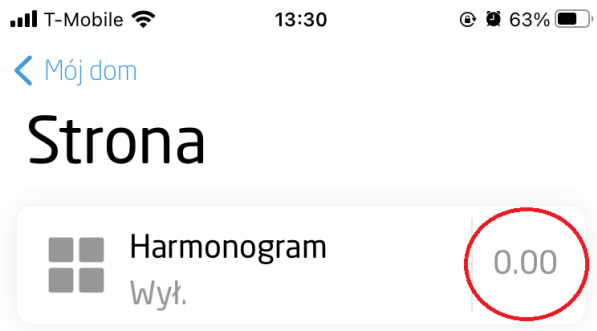
Dla harmonogramów zdefiniowane są gotowe szablony dla widgetu SCHEDULER. Aby dodać widget SCHEDULER z gotowym szablonem należy przeciągnąć obiekt wirtualny harmonogram z listy obiektów na stronę interfejsu:



Skonfigurowany widget SCHEDULER:



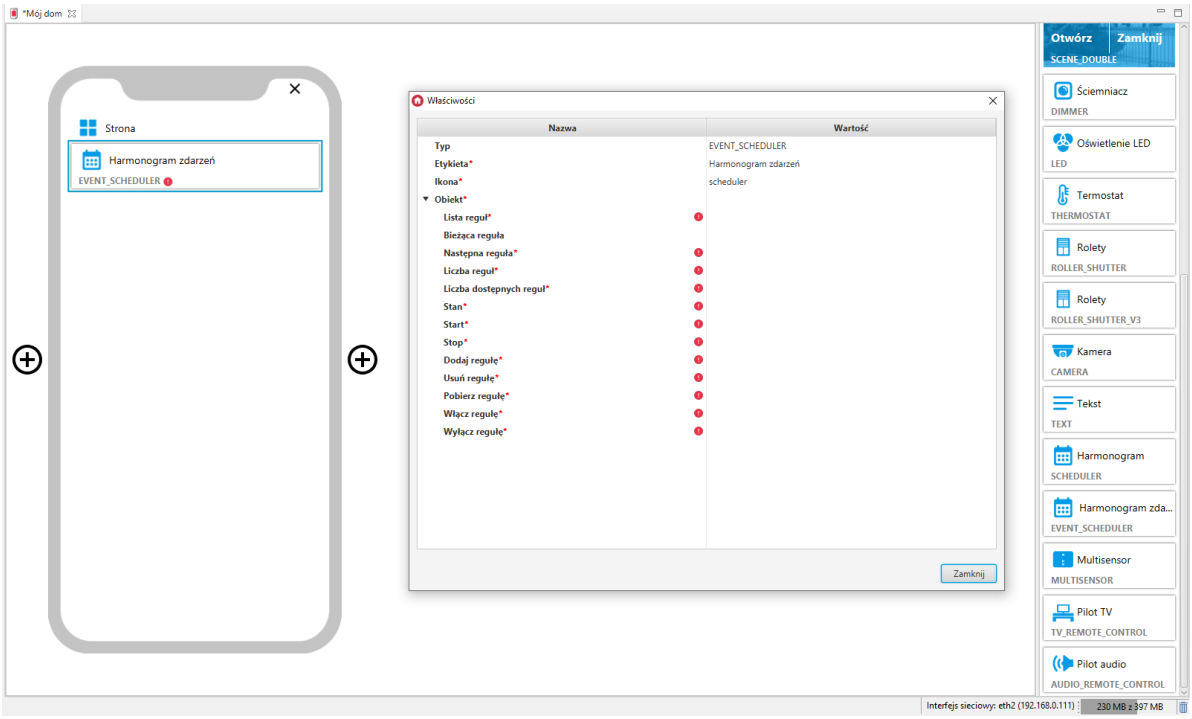
W aplikacji myGrenton istnieje możliwość edycji harmonogramu. Aby to zrobić, należy kliknąć w pole wartości widgetu:



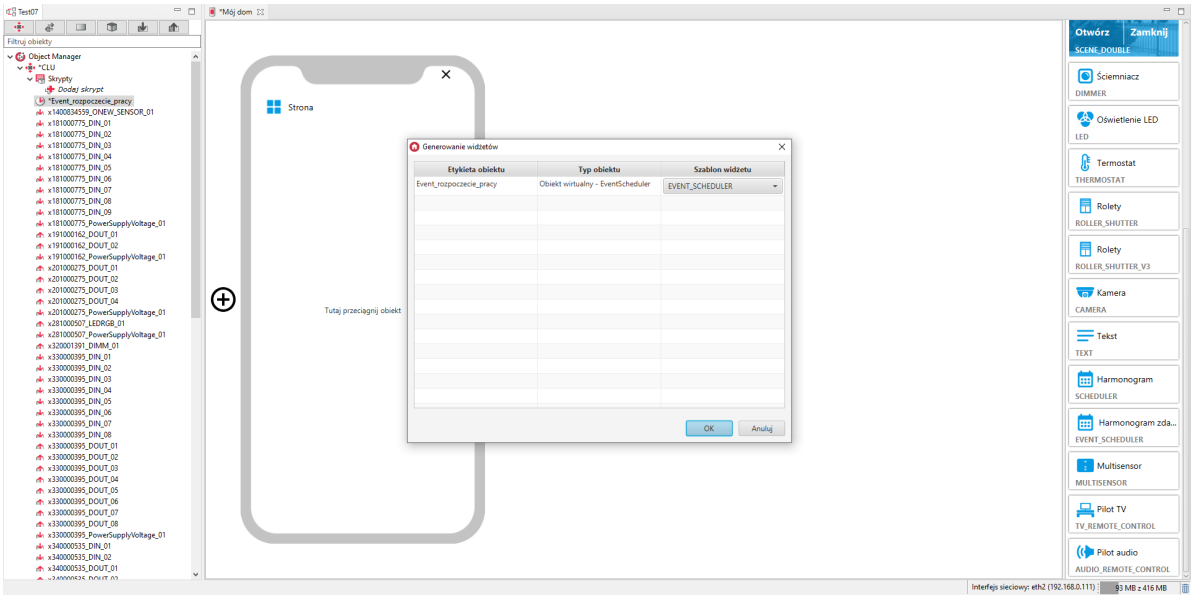
3.19. Harmonogram zdarzeń (EVENT_SCHEDULER)

> **Uwaga!** Widget EVENT_SCHEDULER dostępny jest dla Object Manager w wersji 1.6.0 lub wyższej, CLU w wersji 5.9.1 lub wyższej, dla aplikacji myGrenton w wersji 1.4.0 (Android) / 1.8.0 (iOS) lub wyższej.

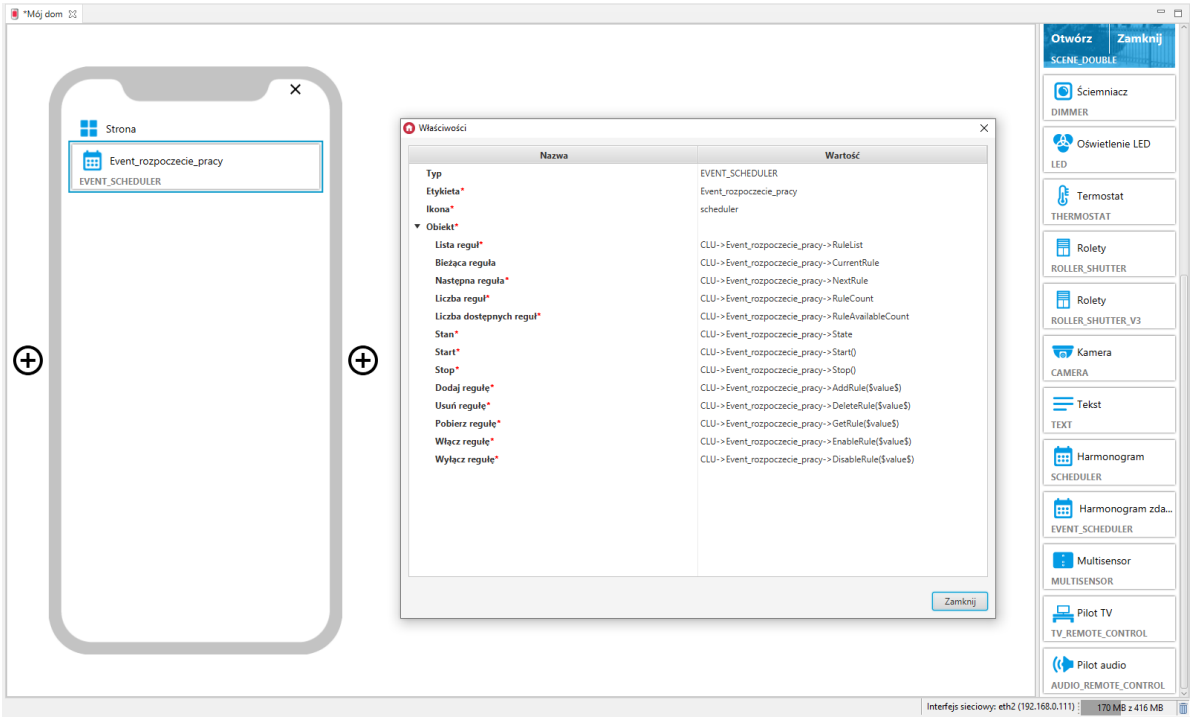
Widget dedykowany dla obiektów wirtualnych typu harmonogram zdarzeń (EventScheduler).



Dla harmonogramów zdarzeń zdefiniowane są gotowe szablony dla widgetu EVENT_SCHEDULER. Aby dodać widget EVENT_SCHEDULER z gotowym szablonem należy przeciągnąć obiekt wirtualny harmonogram zdarzeń z listy obiektów na stronę interfejsu:

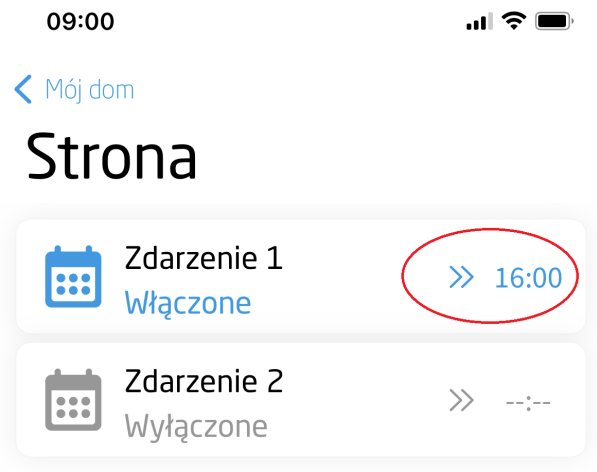


Skonfigurowany widget EVENT_SCHEDULER:

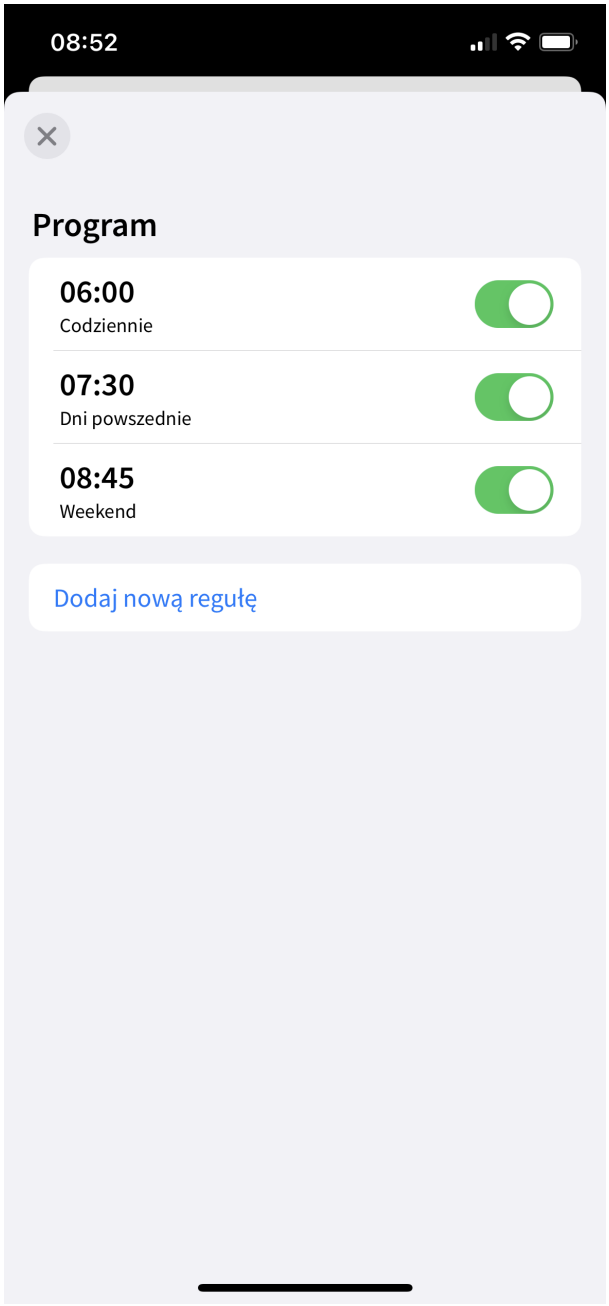


A. Konfiguracja harmonogramu zdarzeń w aplikacji

W aplikacji myGrenton istnieje możliwość edycji reguł harmonogramu zdarzeń. Aby to zrobić, należy kliknąć w pole wartości widgetu:

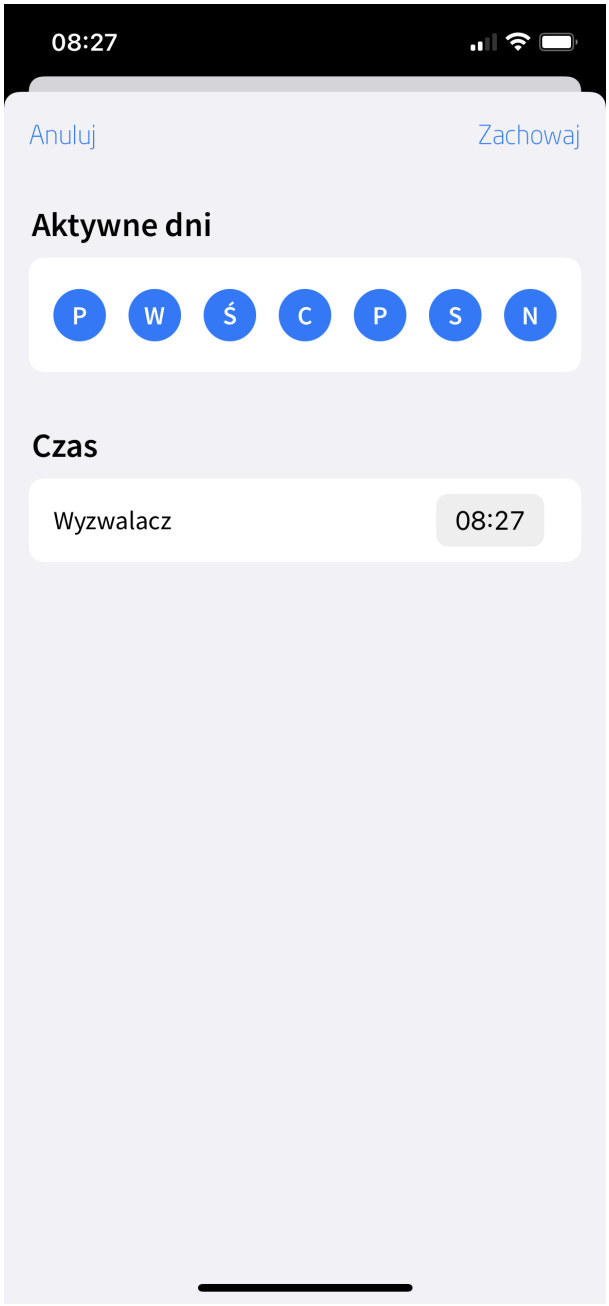


W oknie wyświetlą się reguły pobrane z CLU. Istnieje możliwość edycji tych reguł lub dodania nowych dla poszczególnych dni tygodnia:



Dodawanie nowej reguły

Po wybraniu `Dodaj nową regułę`, otworzy się okno dodawania reguły.

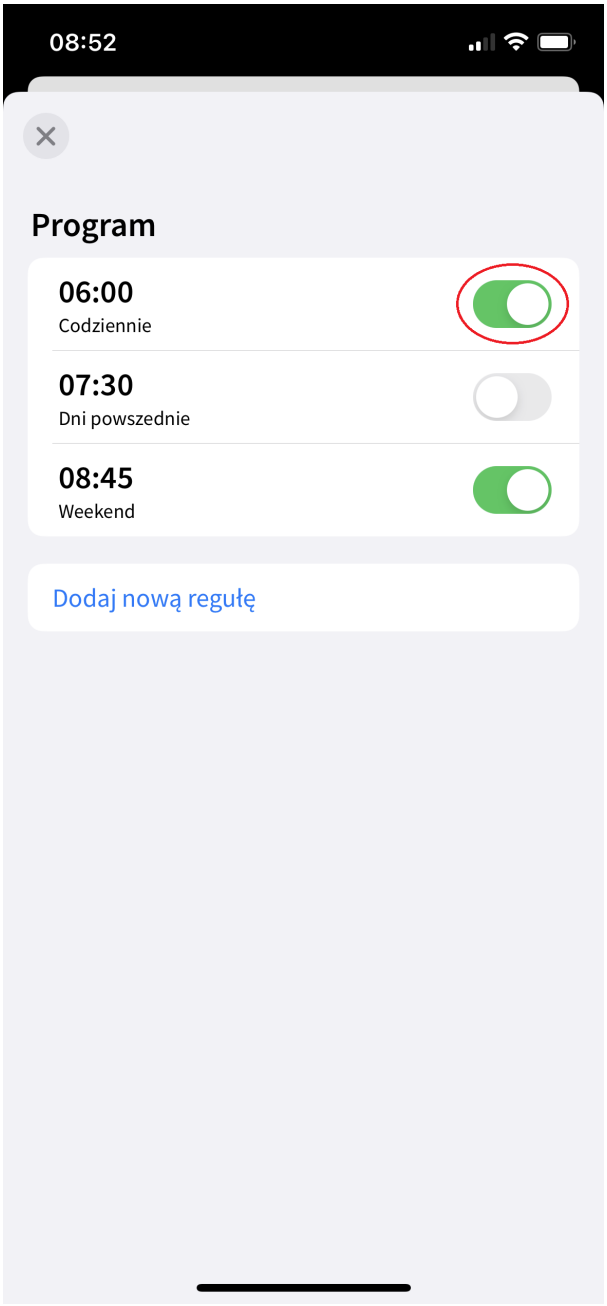


Należy:

- Zaznaczyć wybrane dni tygodnia (co najmniej jeden dzień musi być wybrany),
- Ustawić czas wyzwalacza,
- Akceptować zmiany klikając `Zachowaj`.

Aktywowanie / dezaktywowanie reguły

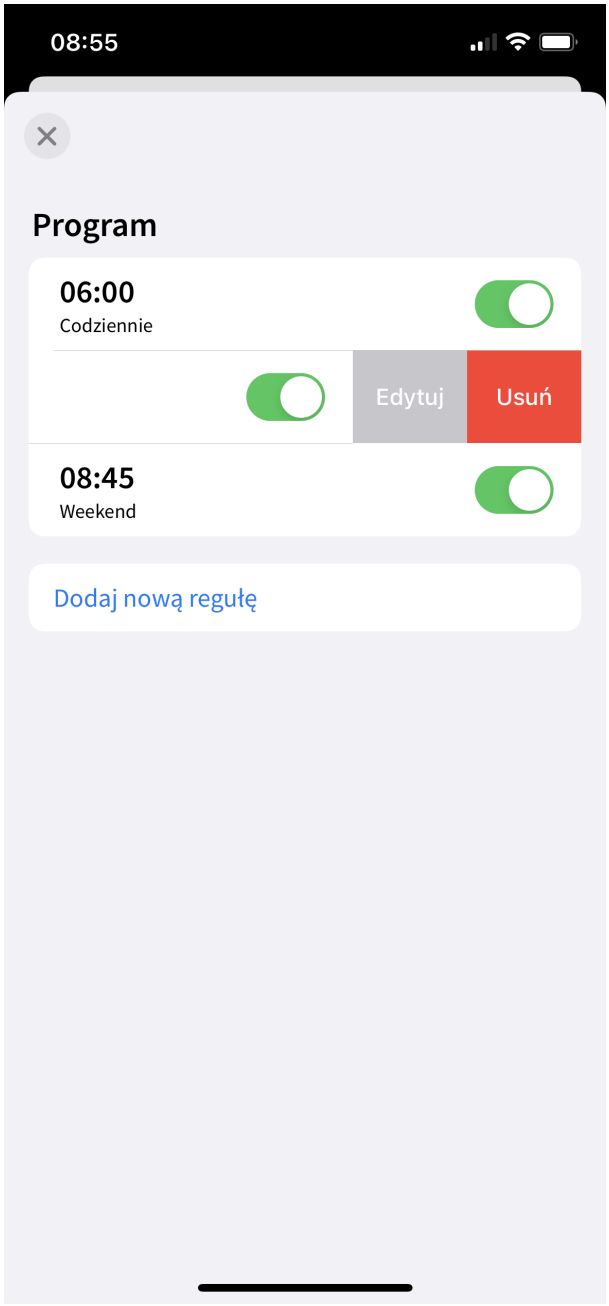
Aby aktywować / dezaktywować regułę należy kliknąć na suwak w prawej części reguły:



Usuwanie / edytowanie reguły

Aby przejść do edycji istniejącej reguły należy kliknąć na regułę (Android) lub wykonać gest przesunięcia w lewo na regule, a następnie kliknąć na opcję `Edytuj` (iOS).

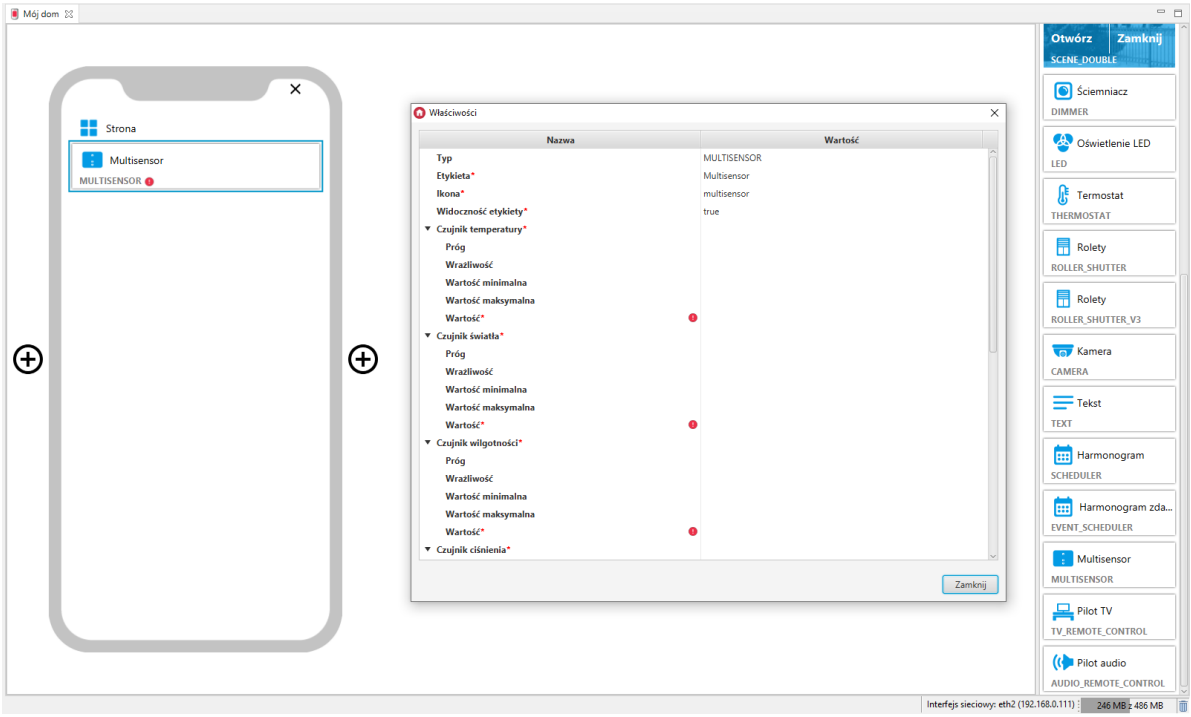
Aby usunąć regułę należy wykonać gest przesunięcia w lewo na regule, a następnie kliknąć opcję `Usuń`.



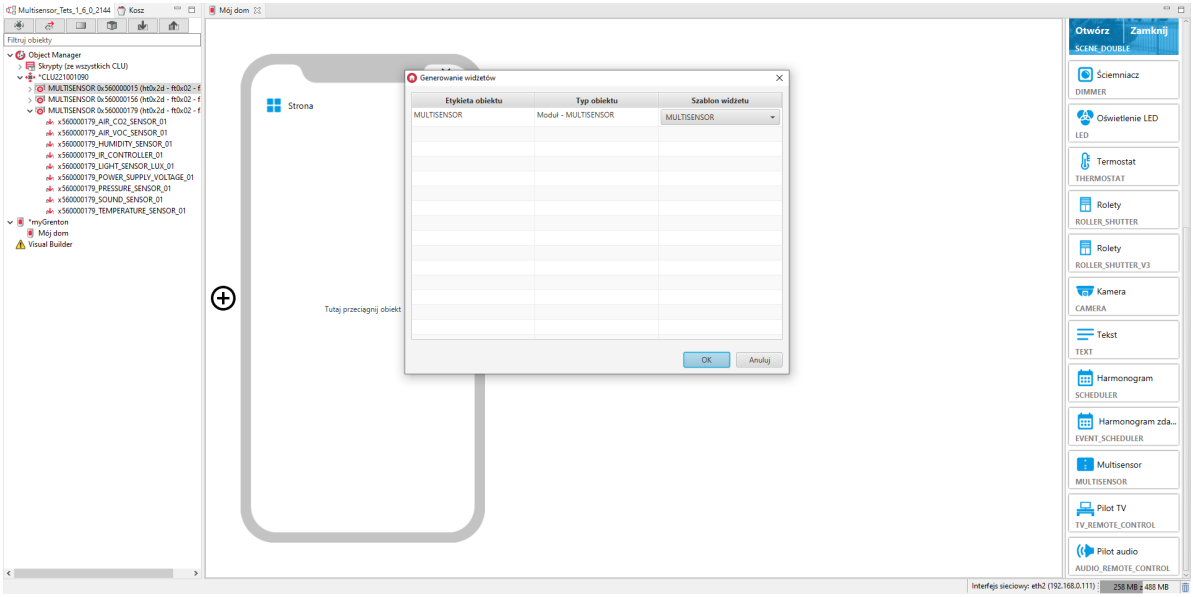
3.20. Multisensor (MULTISENSOR)

> **Uwaga!** Widget MULTISENSOR dostępny jest dla Object Manager w wersji 1.6.0 lub wyższej, CLU w wersji 5.9.1 lub wyższej, modułu Multisensor IR w wersji 1.2.6 lub wyższej, dla aplikacji myGrenton w wersji 1.4.0 (Android) / 1.8.0 (iOS) lub wyższej.

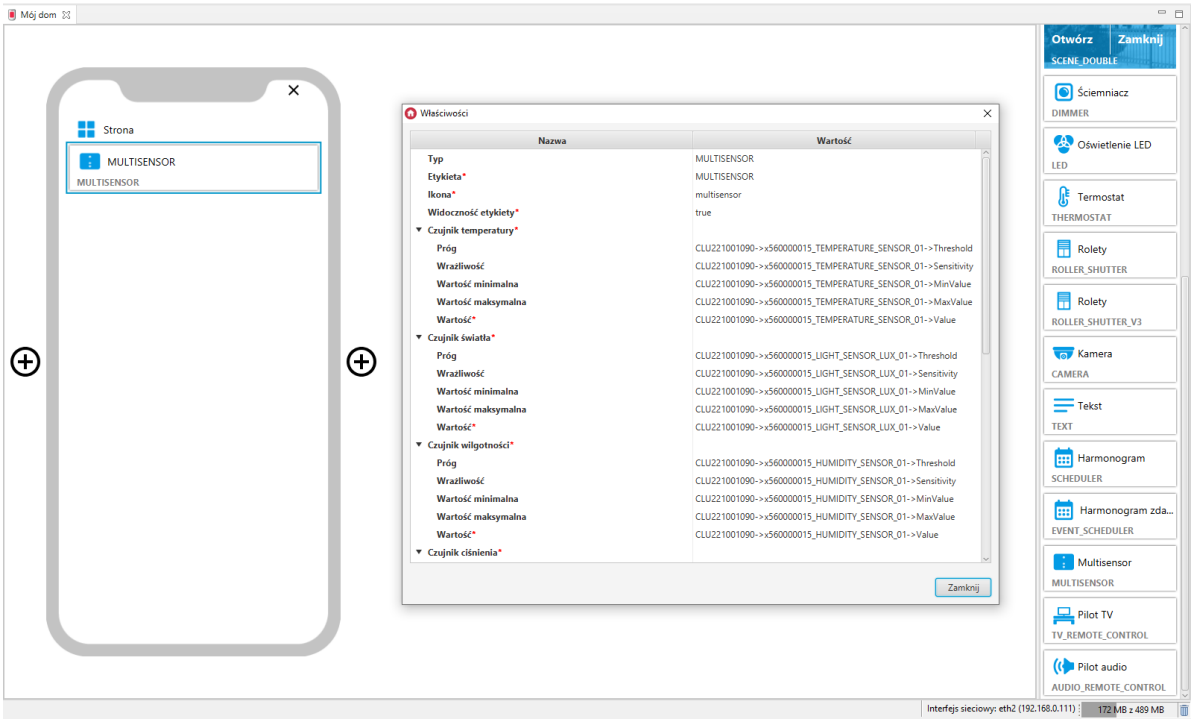
Widget dedykowany do wyświetlania parametrów urządzenia GRENTON MULTISENSOR IR. Pozwala na wyświetlanie parametrów środowiskowych takich jak: temperatura, wilgotność, CO2, LZO (Lotne Związki Organiczne), ciśnienie powietrza, natężenie dźwięku oraz natężenie światła. Widget dysponuje właściwością `Widoczność etykiety`, która umożliwia wyświetlanie lub ukrycie etykiety widgetu w aplikacji. W aplikacji myGrenton możliwa zmiana kolejności wyświetlanych parametrów.



Dla obiektu MULTISENSOR zdefiniowane są gotowe szablony dla widgetu MULTISENSOR. Aby dodać widget MULTISENSOR z gotowym szablonem należy przeciągnąć obiekt MULTISENSOR z listy obiektów na stronę interfejsu:



Skonfigurowany widget MULTISENSOR:



Wygląd widgetu w aplikacji myGrenton, wersje z widoczną oraz ukrytą etykietą:

Mój dom

Strona

Multisensor

Temperatura
24.50°C

Ciśnienie
1013.25 hPa

Światło
22 lx

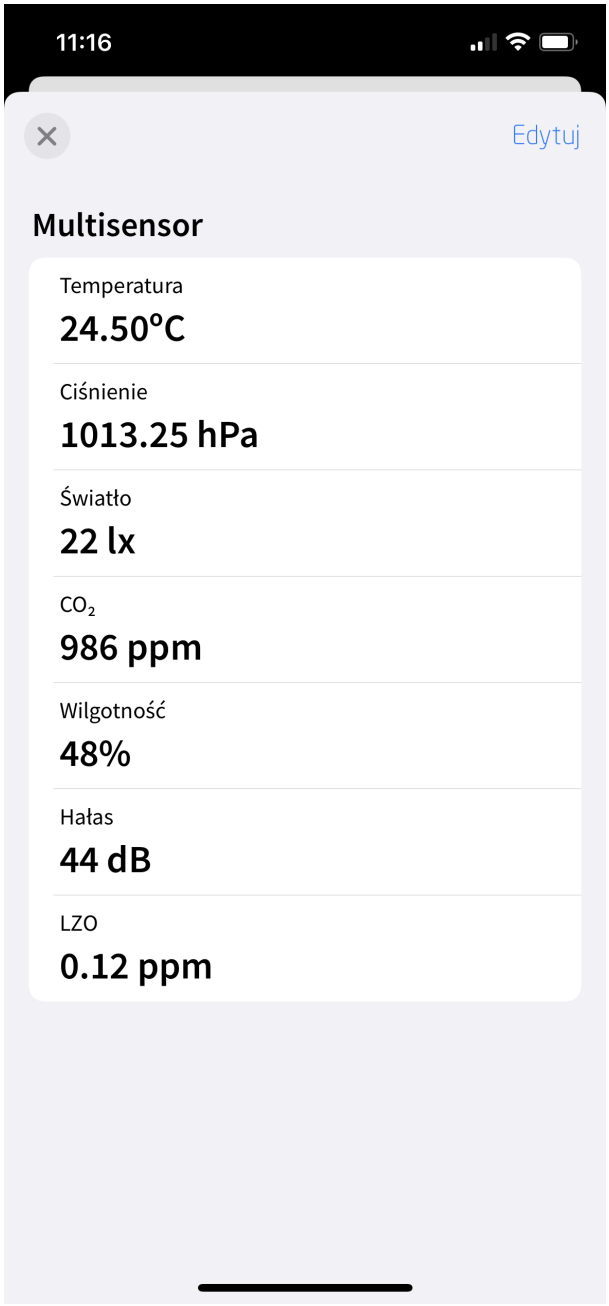
CO_2
986 ppm

Temperatura
24.50°C

Ciśnienie
1013.25 hPa

Światło
22 lx

CO_2
986 ppm



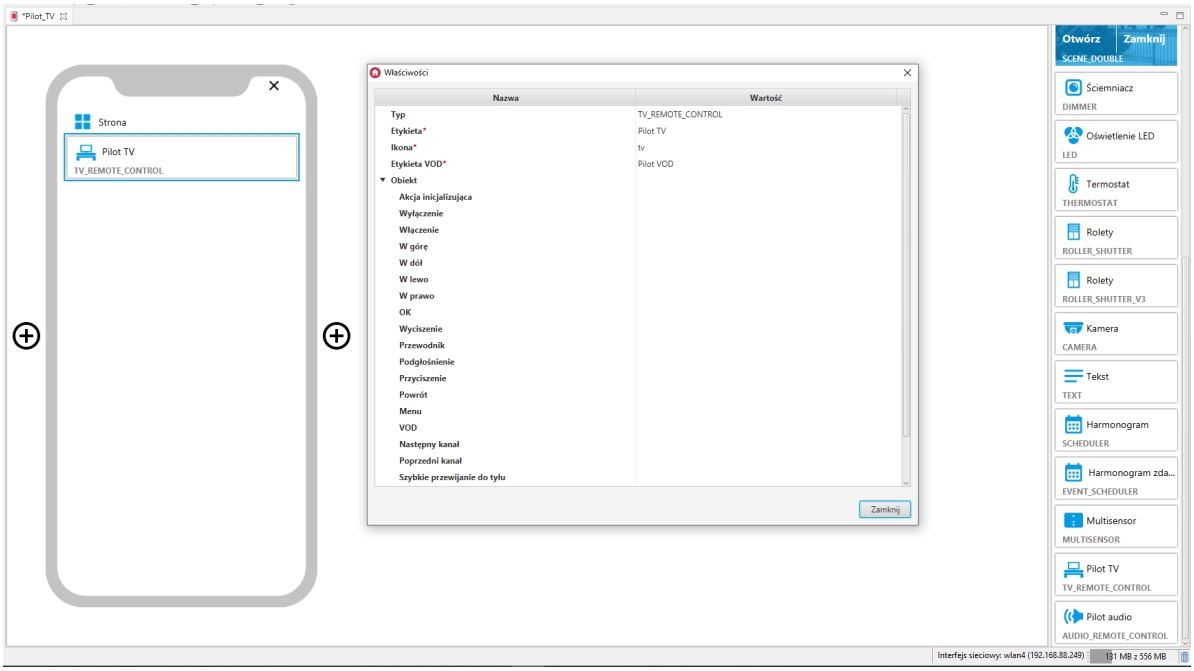
3.21. Pilot TV (TV_REMOTE_CONTROL)

> **Uwaga!** Widget TV_REMOTE_CONTROL dostępny jest dla Object Manager w wersji 1.6.0 lub wyższej, dla aplikacji myGrenton w wersji 1.4.0 (Android) / 1.8.0 (iOS) lub wyższej.

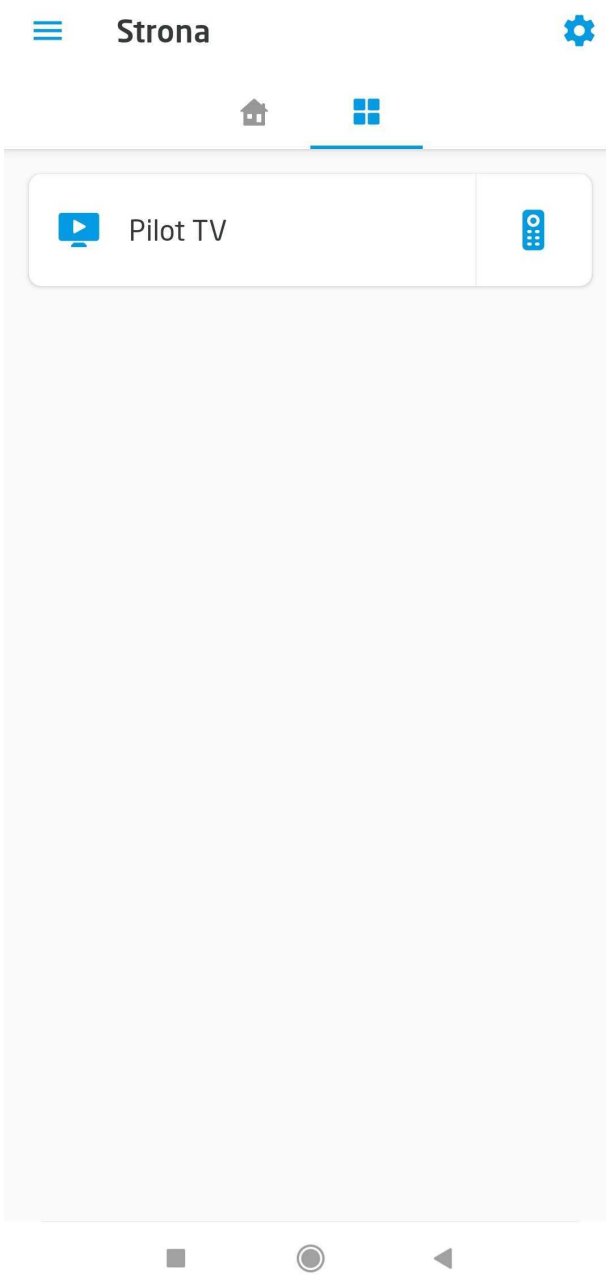
Widget wzorowany na wyglądzie pilota TV, umożliwia pracę z urządzeniami zintegrowanymi z systemem Grenton i zastąpienie tradycyjnego pilota. Sterowanie odbywa się za pomocą akcji przypisanych do wybranych przycisków. Widget TV_REMOTE_CONTROL zawiera:

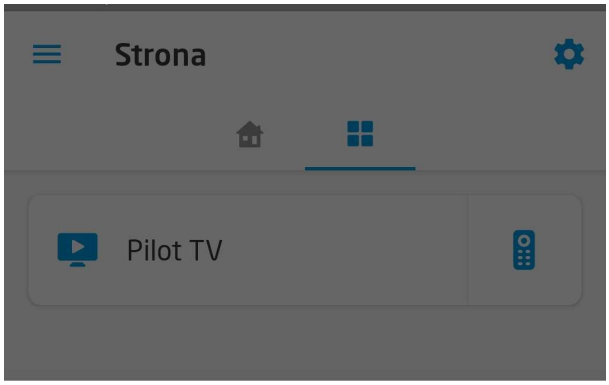
- akcję inicjalizującą - akcja wywoływana w momencie kliknięcia w widget na aplikacji myGrenton (możliwość przypisania dowolnej akcji),
- przyciski do sterowania załączeniem / wyłączeniem urządzenia,
- przyciski do zmiany kanału (programu) oraz zmiany głośności urządzenia,
- przyciski do sterowania odtwarzaniem (pazua, uruchom, przewiń, cofnij),
- przyciski funkcyjne (menu, VOD, powrót, itp.).

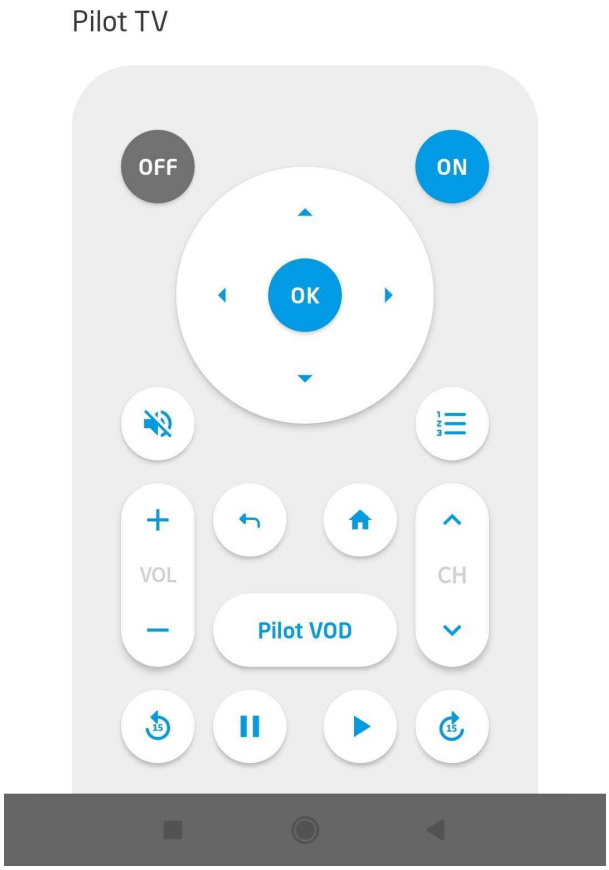
Obsługa w widgecie Pilot TV jest jednokierunkowa, nie ma możliwości odczytu stanu urządzenia.



Wygląd widgetu w aplikacji myGrenton:







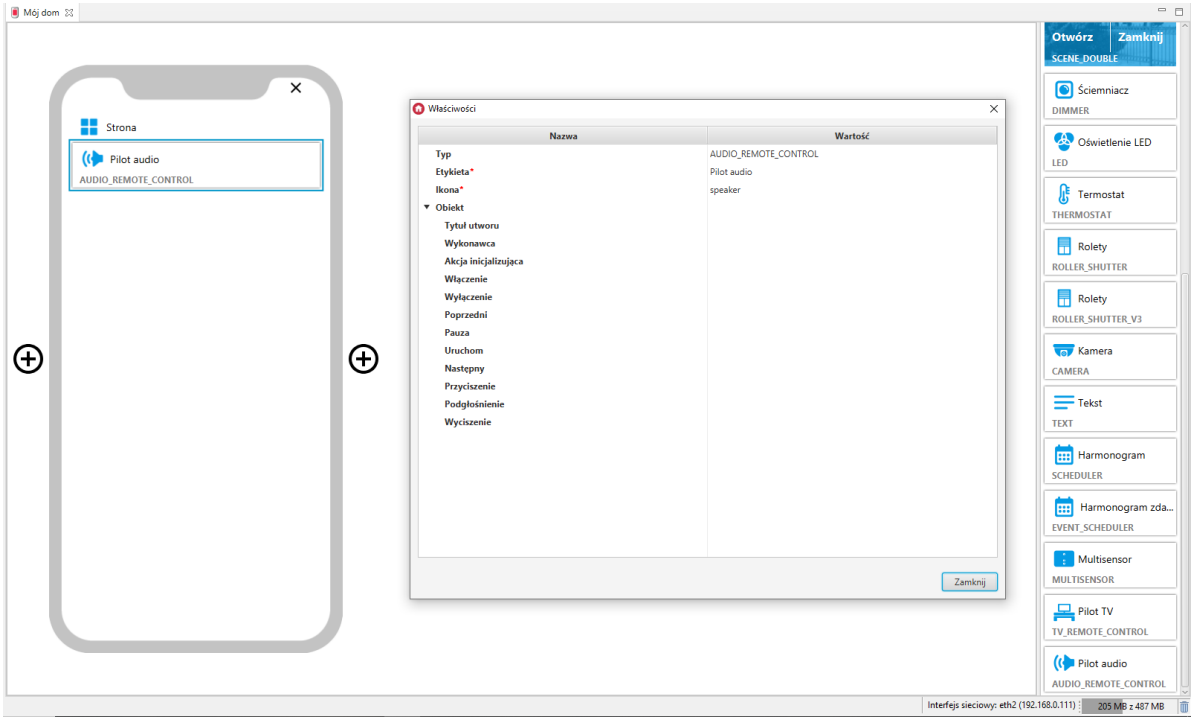
3.22. Pilot Audio (AUDIO_REMOTE_CONTROL)

> **Uwaga!** Widget AUDIO_REMOTE_CONTROL dostępny jest dla Object Manager w wersji 1.6.0 lub wyższej, dla aplikacji myGrenton w wersji 1.4.0 (Android) / 1.8.0 (iOS) lub wyższej.

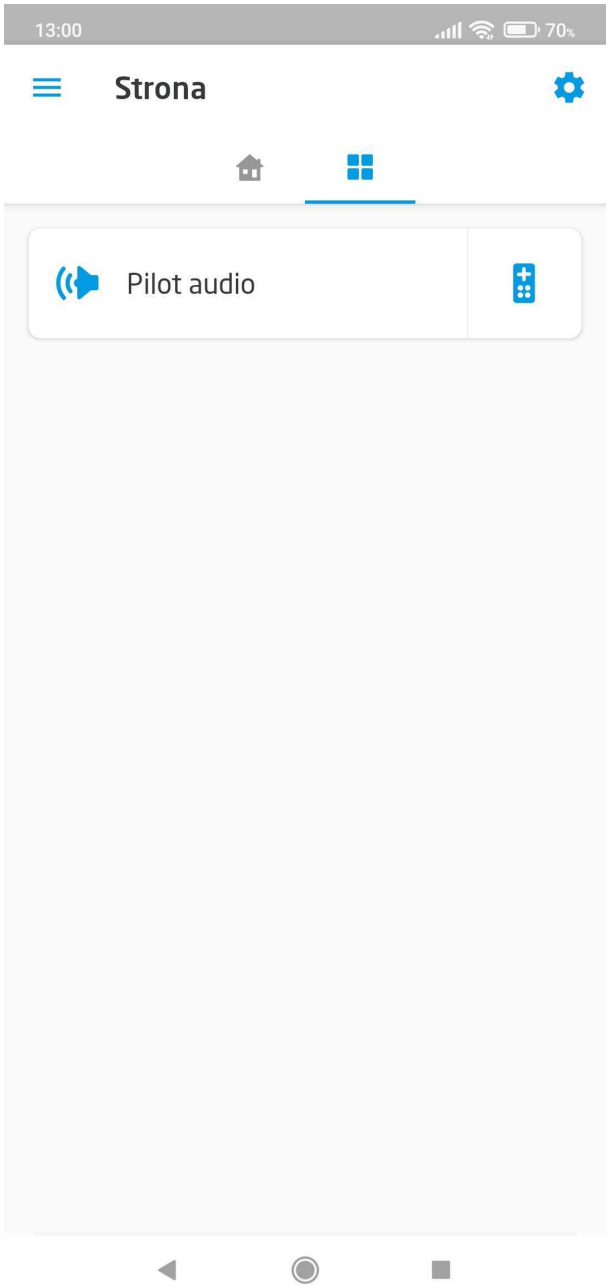
Widget dedykowany do sterowania systemami audio, umożliwia pracę z urządzeniami zintegrowanymi z systemem Grenton. Sterowanie odbywa się za pomocą akcji przypisanych do wybranych przycisków. Widget AUDIO_REMOTE_CONTROL zawiera:

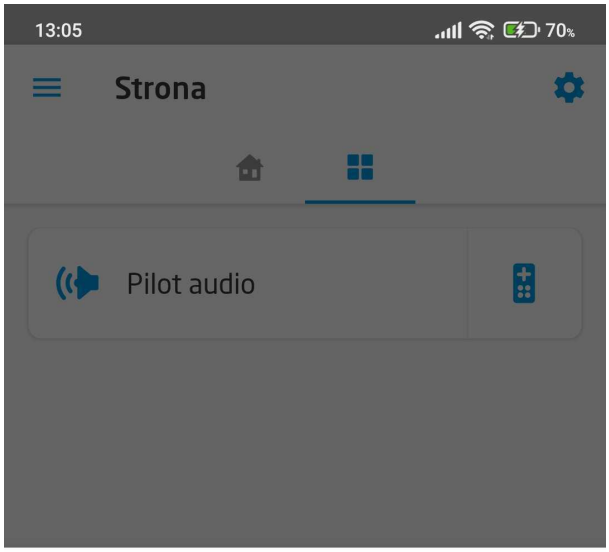
- akcję inicjalizującą - akcja wywoływana w momencie kliknięcia w widget na aplikacji myGrenton (możliwość przypisania dowolnej akcji),
- przyciski do sterowania załączeniem / wyłączeniem urządzenia,
- przyciski do zmiany głośności urządzenia,
- przyciski do sterowania odtwarzaniem (pazua, uruchom, następny, poprzedni),
- pola wyświetlające wykonawcę oraz tytuł.

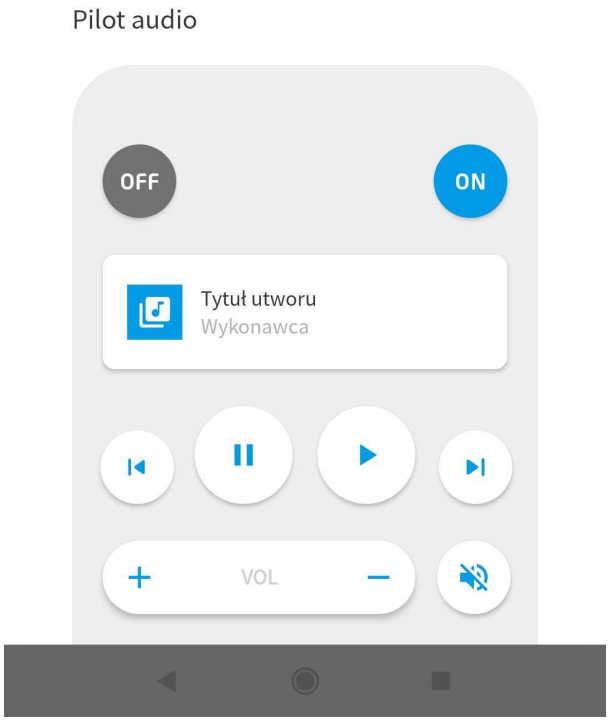
Obsługa w widgecie Pilot Audio jest jednokierunkowa, nie ma możliwości odczytu stanu urządzenia. Możliwe jest pobranie informacji z wybranej cechy (np. użytkownika) i wyświetlenie jej w polu Wykonawca i/lub Tytuł.



Wygląd widgetu w aplikacji myGrenton:


13:00
70%
Strona
Pilot audio



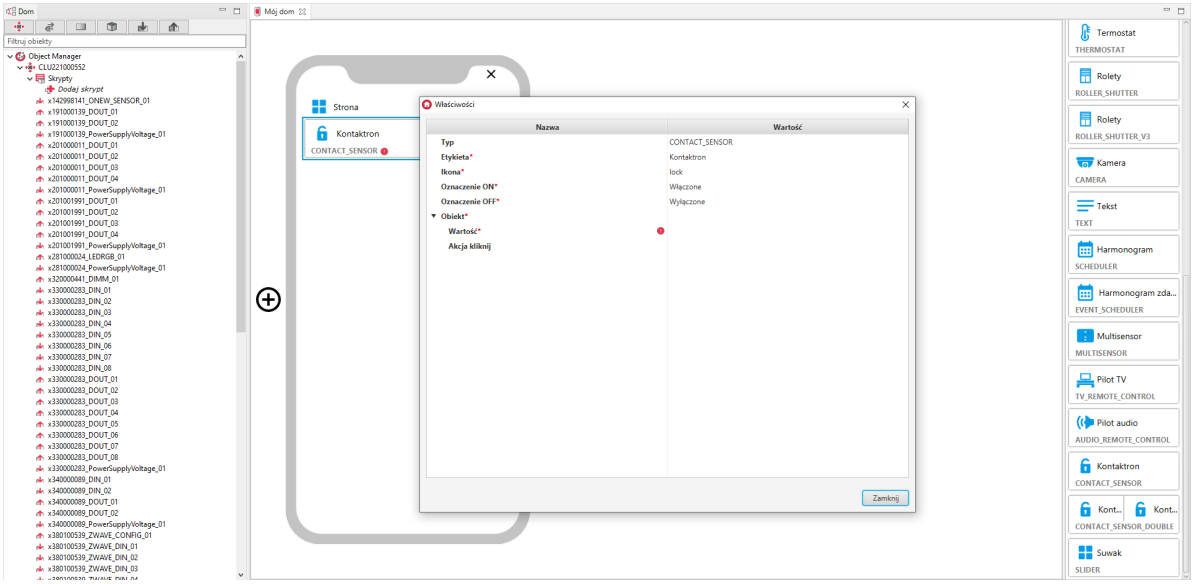


3.22. Kontaktron (CONTACT_SENSOR)

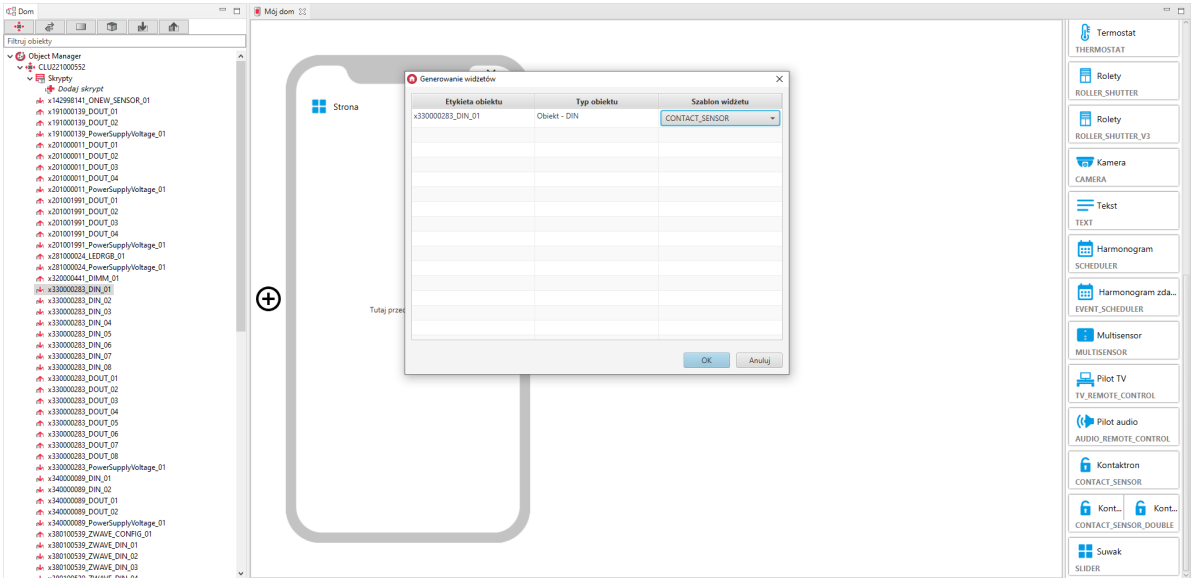
> **Uwaga!** Widget CONTACT_SENSOR dostępny jest dla Object Manager w wersji 1.7.0 lub wyższej, dla aplikacji myGrenton w wersji 1.5.0 (Android) / 1.9.0 (iOS) lub wyższej.

Widget dedykowany do wyświetlania stanów Włączone / Wyłączone na podstawie wartości obiektów, cech. Widget CONTACT_SENSOR zawiera:

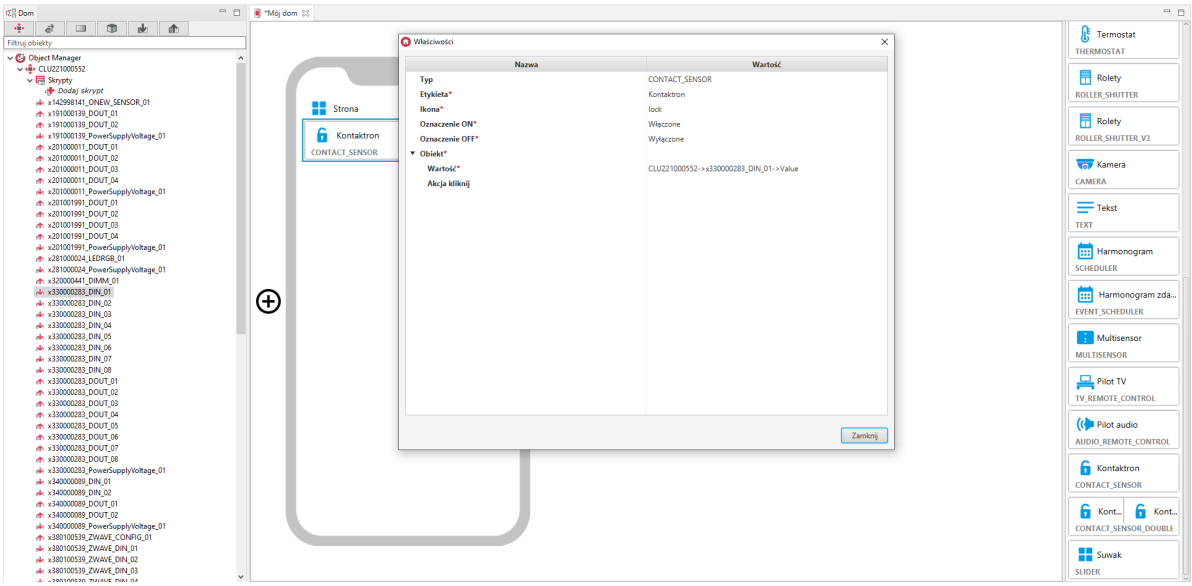
- obsługa stanów Włączone / Wyłączone - na podstawie wartości obiektów, cech (`0/1` lub `true/false`) odpowiednio wyświetlany stan widgetu,
- oznaczenia - możliwość ustawienia dowolnej nazwy stanów Włączone / Wyłączone poprzez właściwości `Oznaczenie ON`, `Oznaczenie OFF`,
- opcjonalna akcja kliknij - akcja wywoływana w momencie kliknięcia na widget (możliwość przypisania dowolnej akcji).



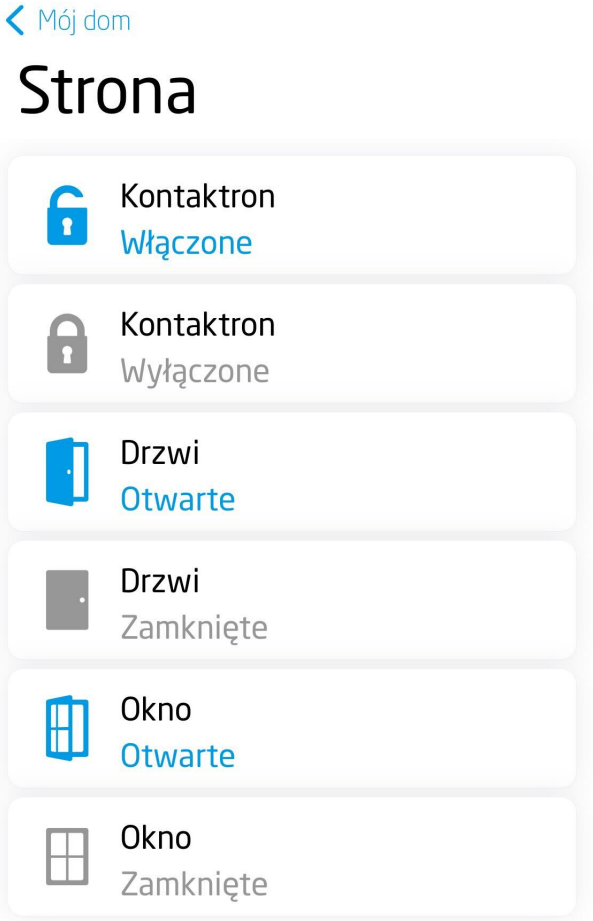
Dla obiektów DIN zdefiniowane są gotowe szablony dla widgetu CONTACT_SENSOR. Aby dodać widget CONTACT_SENSOR z gotowym szablonem należy przeciągnąć obiekt z listy obiektów na stronę interfejsu:



Skonfigurowany widget CONTACT_SENSOR:



Wygląd widgetu w aplikacji myGrenton:

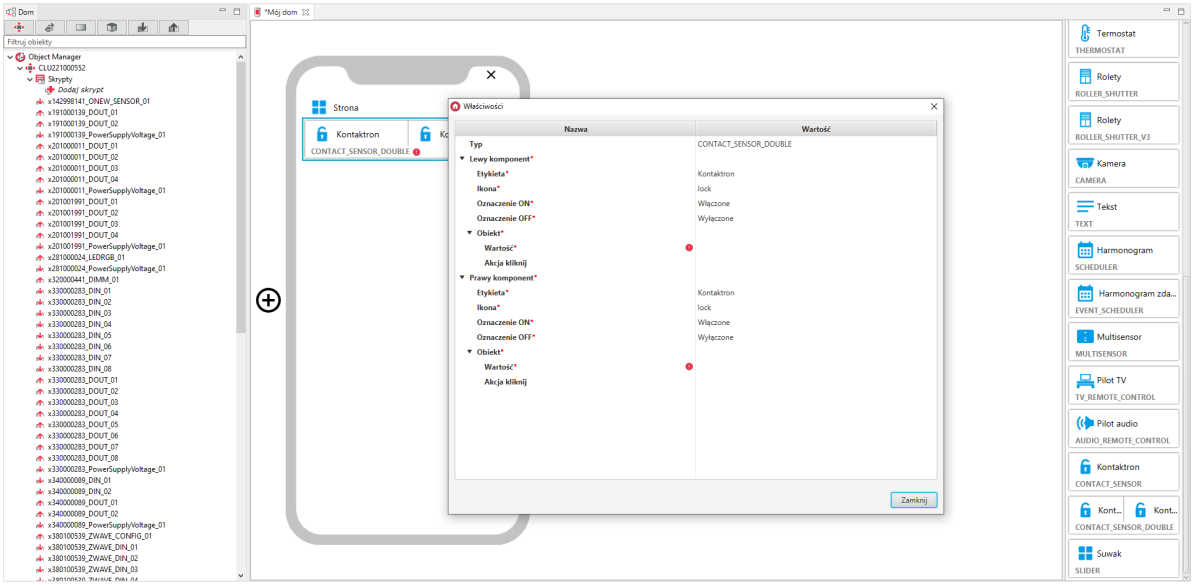


3.23. Kontaktron Podwójny (CONTACT_SENSOR_DOUBLE)

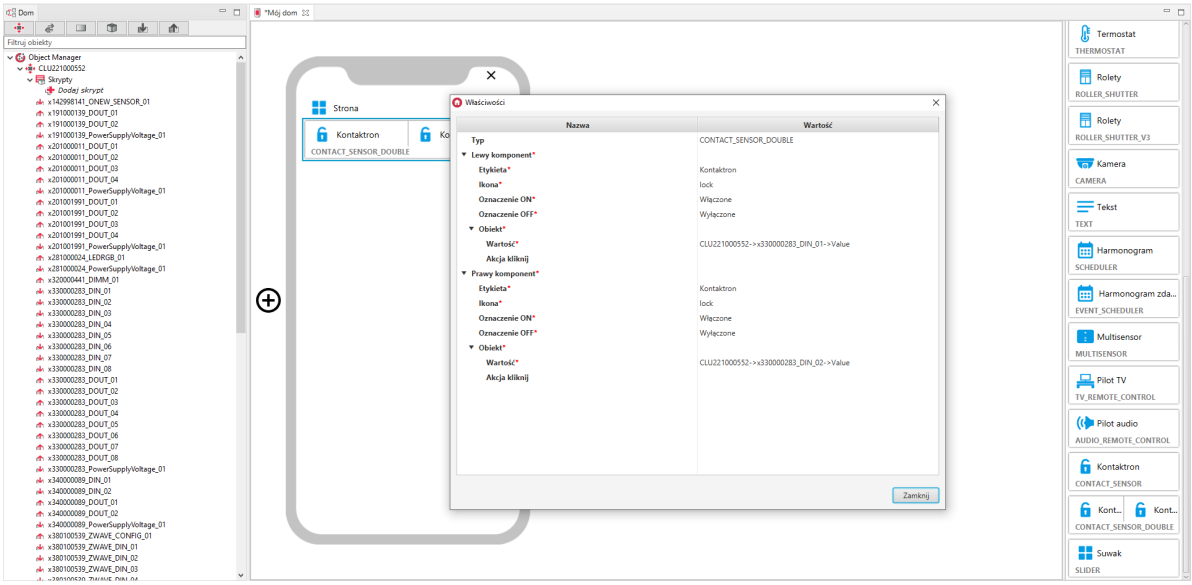
> **Uwaga!** Widget CONTACT_SENSOR_DOUBLE dostępny jest dla Object Manager w wersji 1.7.0 lub wyższej, dla aplikacji myGrenton w wersji 1.5.0 (Android) / 1.9.0 (iOS) lub wyższej.

Jest to podwójna wersja widgetu Kontaktron. Widget dedykowany do wyświetlania stanów Włączone / Wyłączone na podstawie wartości obiektów, cech. Widget CONTACT_SENSOR_DOUBLE zawiera:

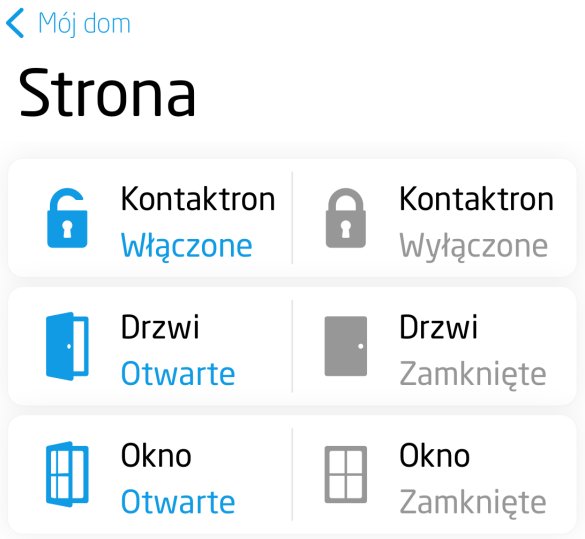
- obsługa stanów Włączone / Wyłączone - na podstawie wartości obiektów, cech (`0/1` lub `true/false`) odpowiednio wyświetlany stan widgetu,
- oznaczenia - możliwość ustawienia dowolnej nazwy stanów Włączone / Wyłączone poprzez właściwości `Oznaczenie ON`, `Oznaczenie OFF`,
- opcjonalna akcja kliknij - akcja wywoływana w momencie kliknięcia na widget (możliwość przypisania dowolnej akcji).



Skonfigurowany widget CONTACT_SENSOR_DOUBLE:



Wygląd widgetu w aplikacji myGrenton:

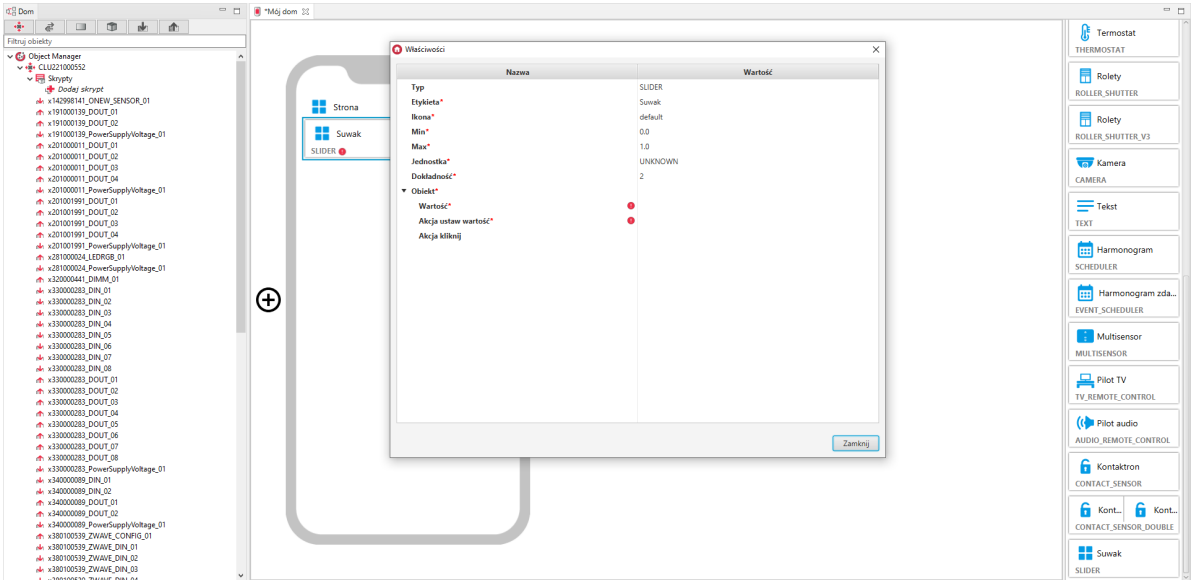


3.24. Suwak (SLIDER)

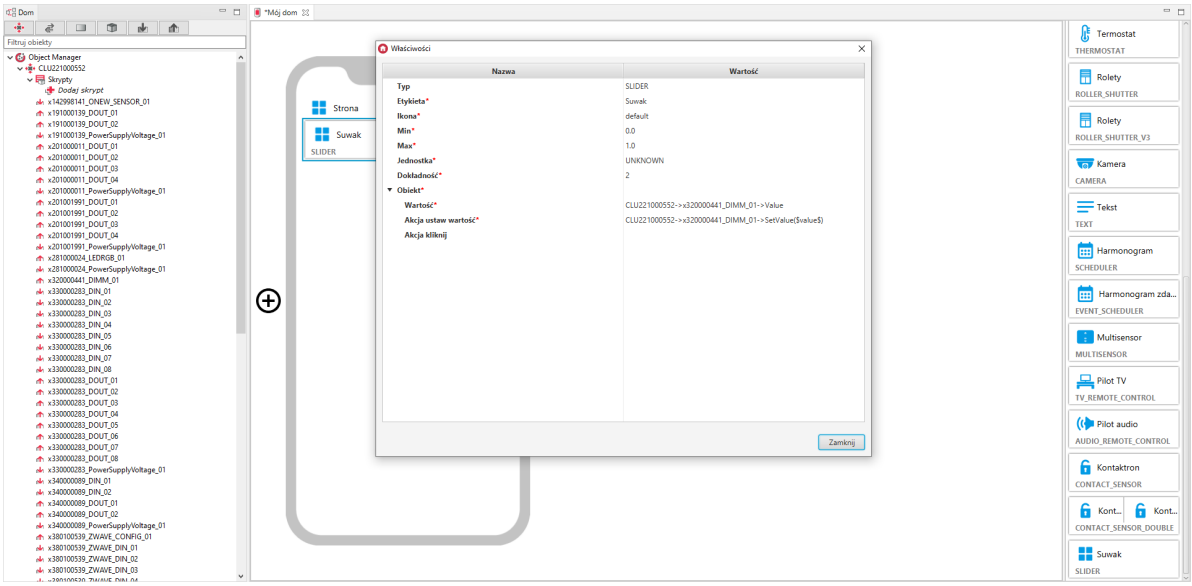
> **Uwaga!** Widget SLIDER dostępny jest dla Object Manager w wersji 1.7.0 lub wyższej, dla aplikacji myGrenton w wersji 1.5.0 (Android) / 1.9.0 (iOS) lub wyższej.

Widget dedykowany do płynnego sterowania wartością. Widget SLIDER zawiera:

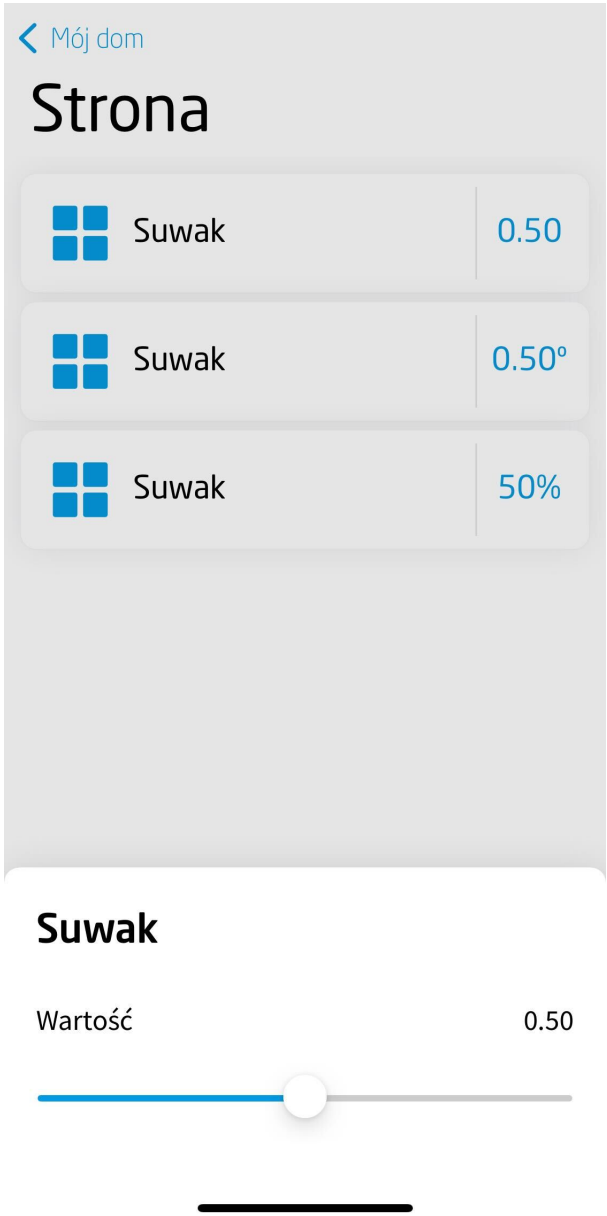
- jednostka - możliwość określenia jednostek `UNKNOWN`, `PERCENT`, `DEGREE`,
- wartość - aktualna wartość wyjścia wyświetlana w prawej części widgetu. Dla jednostki `PERCENT` obliczana na podstawie ustawionych właściwości `Min`, `Max`, dla jednostek `UNKNOWN`, `DEGREE` wartość wyświetlana z liczbą miejsc po przecinku określoną w właściwości `Dokładność`,
- suwak sterujący - wartość ustawiana na podstawie podanego zakresu (właściwości `Min`, `Max`) oraz dokładności (właściwość `Dokładność` określa ilość miejsc po przecinku ustawianej wartości),
- opcjonalna akcja kliknij - akcja wywoływana w momencie kliknięcia na widget (możliwość przypisania dowolnej akcji).



Skonfigurowany widget SLIDER:



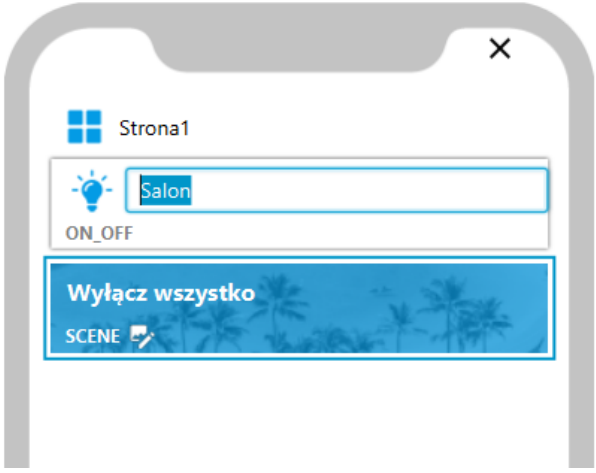
Wygląd widgetu w aplikacji myGrenton:



3.25. Personalizacja widgetu

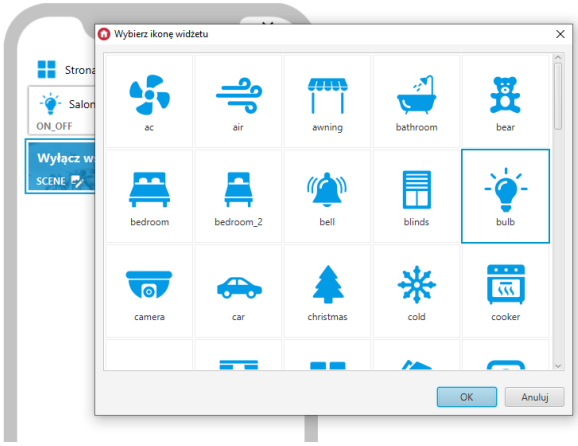
A. Zmiana nazwy widgetu

Zmiana nazwy widgetu możliwa jest między innymi poprzez kliknięcie w aktualną nazwę widgetu. Aby zatwierdzić zmianę należy wcisnąć na klawiaturze *Enter*. Aby anulować, należy wcisnąć na klawiaturze *Esc*.

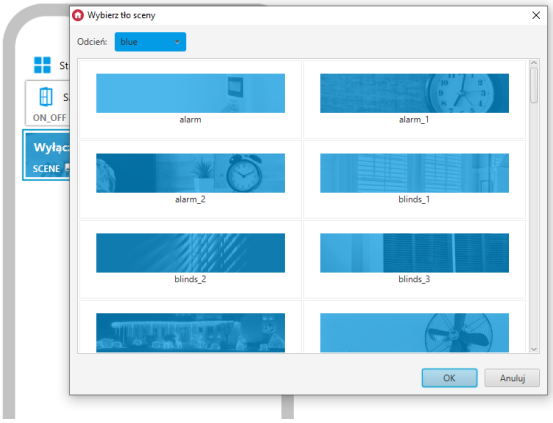


B. Zmiana ikony/obrazka widgetu

Zmiana ikony możliwa jest między innymi poprzez kliknięcie w ikonę obok nazwy widgetu. Wówczas pojawi się okno z dostępnymi ikonami.



Zmiana obrazka tła widgetu Scena lub Otwórz/Zamknij możliwa jest między innymi poprzez kliknięcie w ikonę edycji znajdującą się pod nazwą widgetu. Dodatkowo możliwe jest wybranie odcienia obrazka z dostępnych 15 kolorów.



> **Uwaga!** W przypadku, gdy dla wybranego widgetu zostanie ustawiona ikona lub obrazek tła nieobsługiwany w danej wersji aplikacji myGrenton, zostanie on zastąpiony domyślną ikoną / obrazkiem tła.

3.26. Usuwanie widgetu

Aby usunąć widget należy go zaznaczyć i wcisnąć klawisz `Delete` lub kliknąć prawym przyciskiem myszy i wybrać opcję `Usuń` z menu kontekstowego. Na urządzeniu z zainstalowanym systemem operacyjnym macOS należy użyć kombinacji znaków `Fn`+`Backspace`.

3.27. Kopiowanie widgetu

Aby skopiować widget należy go zaznaczyć i wcisnąć kombinację klawiszy `Ctrl`+`C` (następnie `Ctrl`+`V` w miejscu docelowym) lub wybrać opcję `Kopiuj` z menu kontekstowego (następnie PPM i `Wklej` w miejscu docelowym).

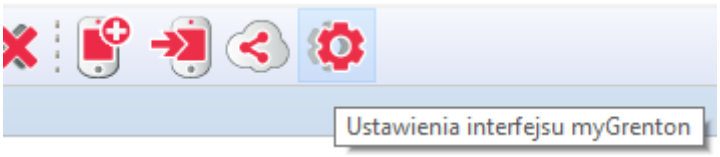
Możliwe jest zaznaczenie większej ilości widgetów do skopiowania. Można to zrobić poprzez zaznaczanie przy jednocześnie wciśniętym klawiszu `Ctrl`, lub poprzez zaznaczenie przedziału przy jednocześnie wciśniętym klawiszu `Shift`.

Widgety mogą zostać skopiowane na wszystkie strony interfejsu, również na strony innych interfejsów w ramach jednego projektu.

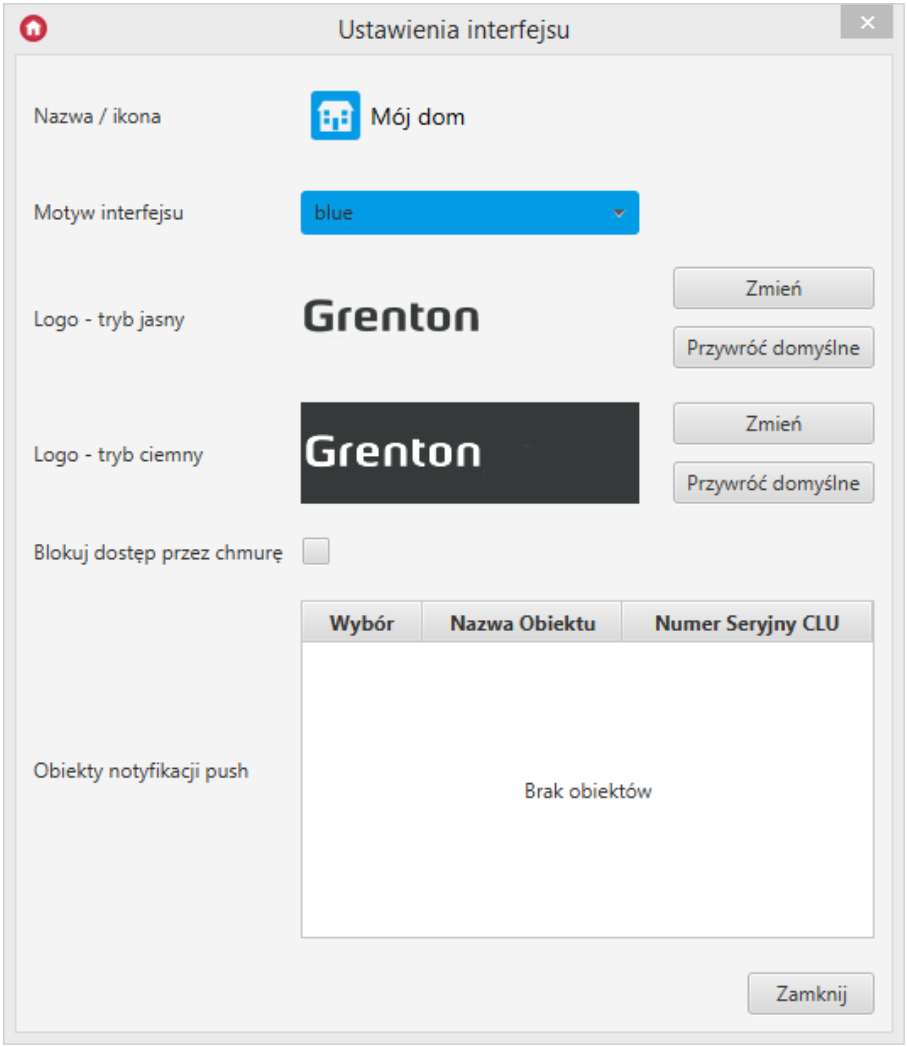
Uwaga! Jeśli widgety zostaną skopiowane do innego projektu, konieczna będzie ich ponowna konfiguracja.

4. Personalizacja interfejsu

Istnieje możliwość dostosowania wyglądu interfejsu do swoich preferencji. Aby spersonalizować interfejs należy kliknąć ikonę `Ustawienia interfejsu myGrenton` w pasku narzędzi:

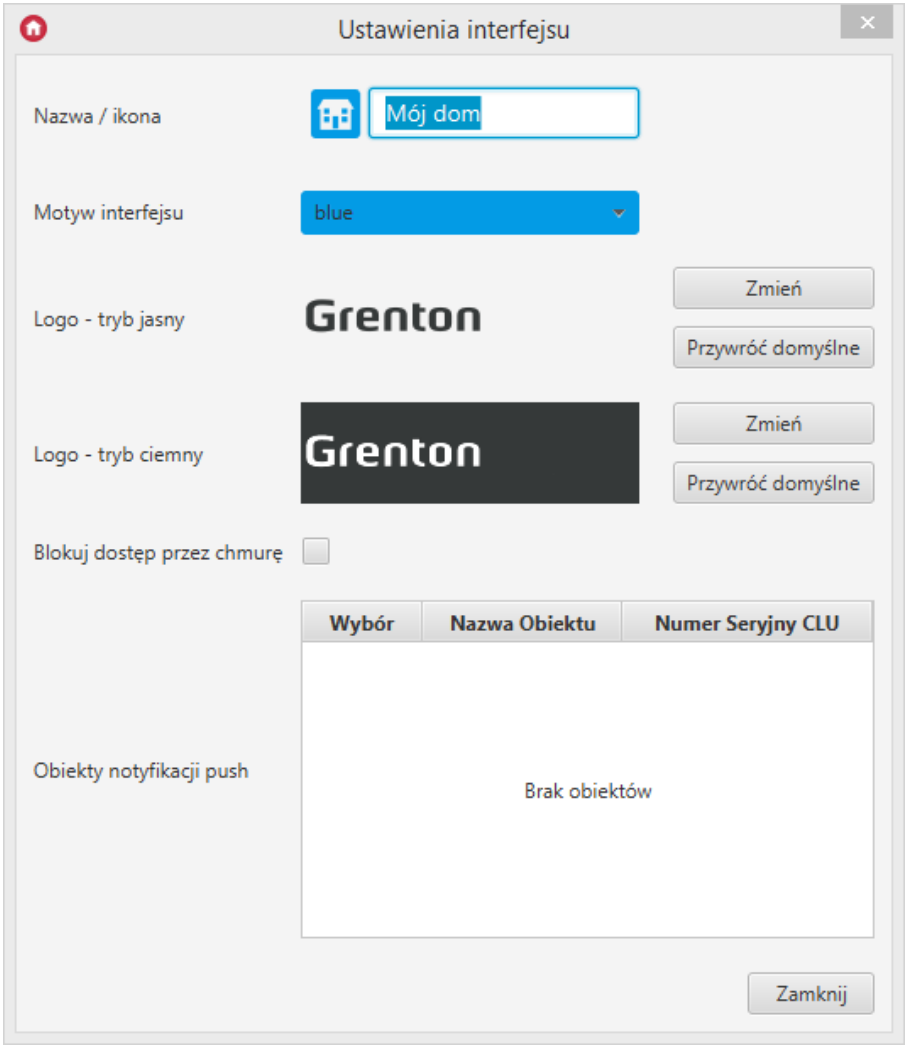


Po kliknięciu ikony zębatki pojawi się okno z ustawieniami interfejsu:



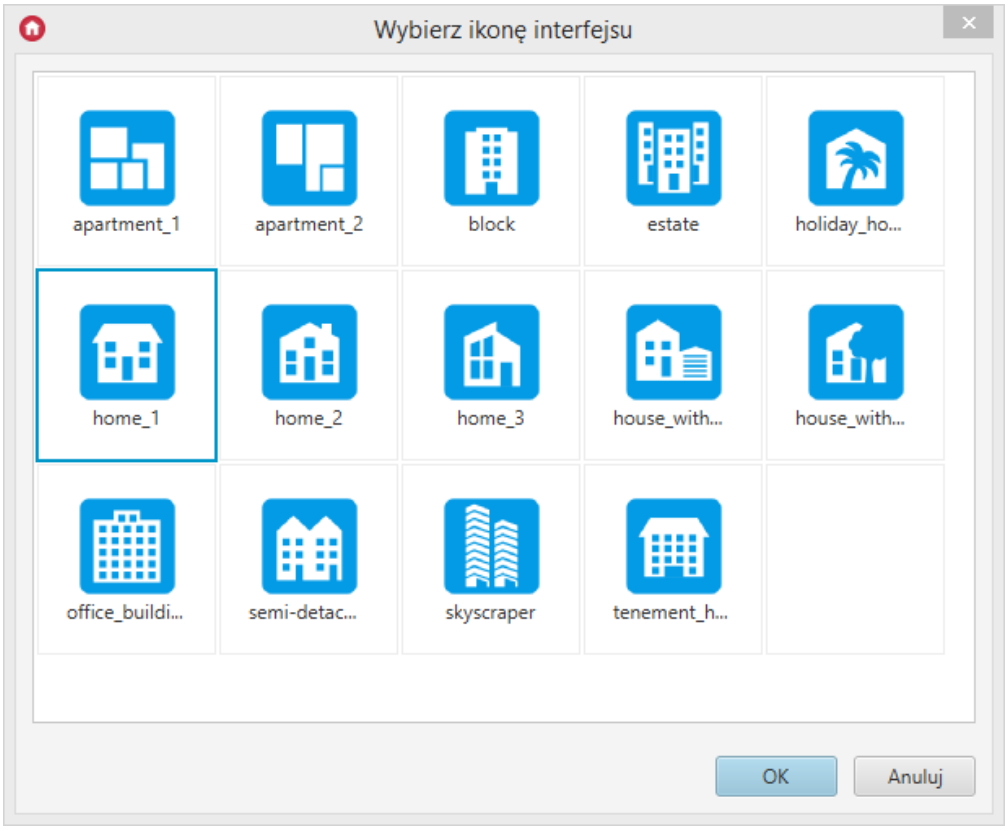
4.1. Zmiana nazwy interfejsu

Aby zmienić nazwę interfejsu należy kliknąć na obecną nazwę wyświetlaną w górnej części okna. Aby zatwierdzić zmianę należy wcisnąć na klawiaturze *Enter*. Aby anulować, należy wcisnąć na klawiaturze *Esc*.



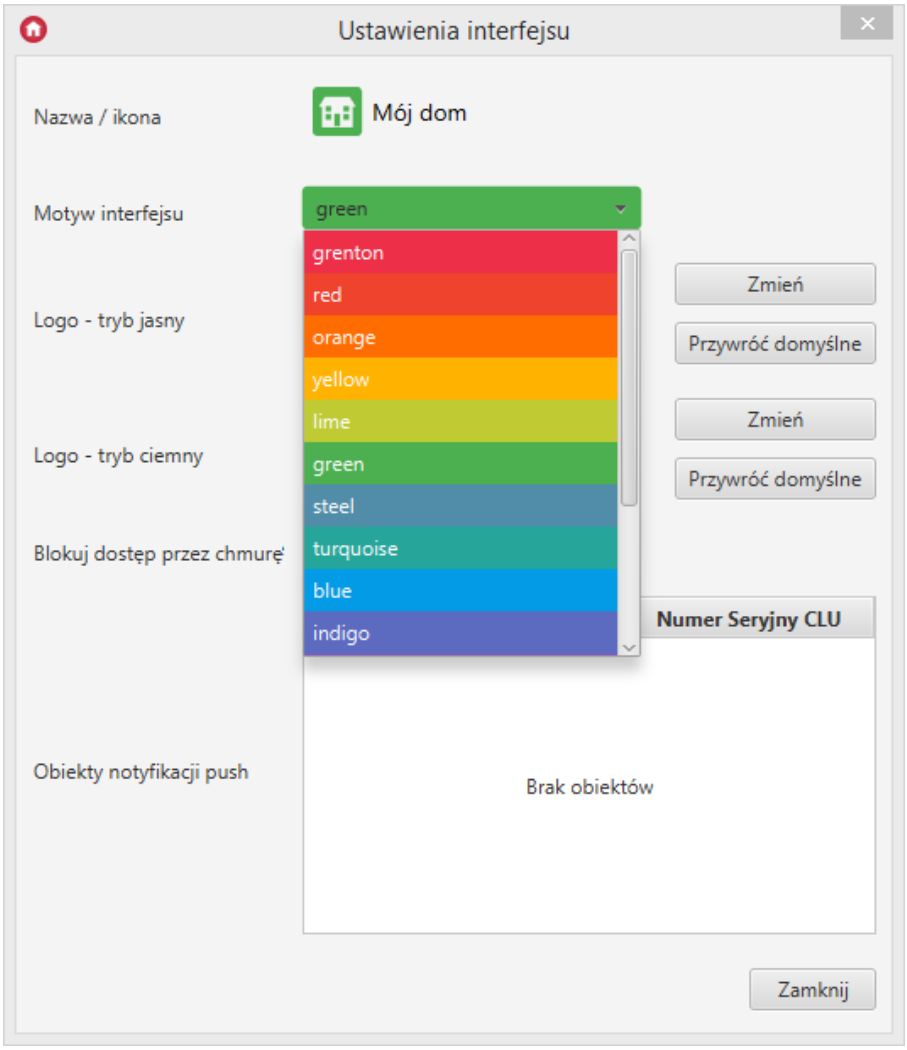
4.2. Zmiana ikony interfejsu

Aby zmienić ikonę interfejsu należy kliknąć w obrazek ikony znajdujący się obok nazwy interfejsu. Otworzy się wówczas okno z dostępnymi ikonami.



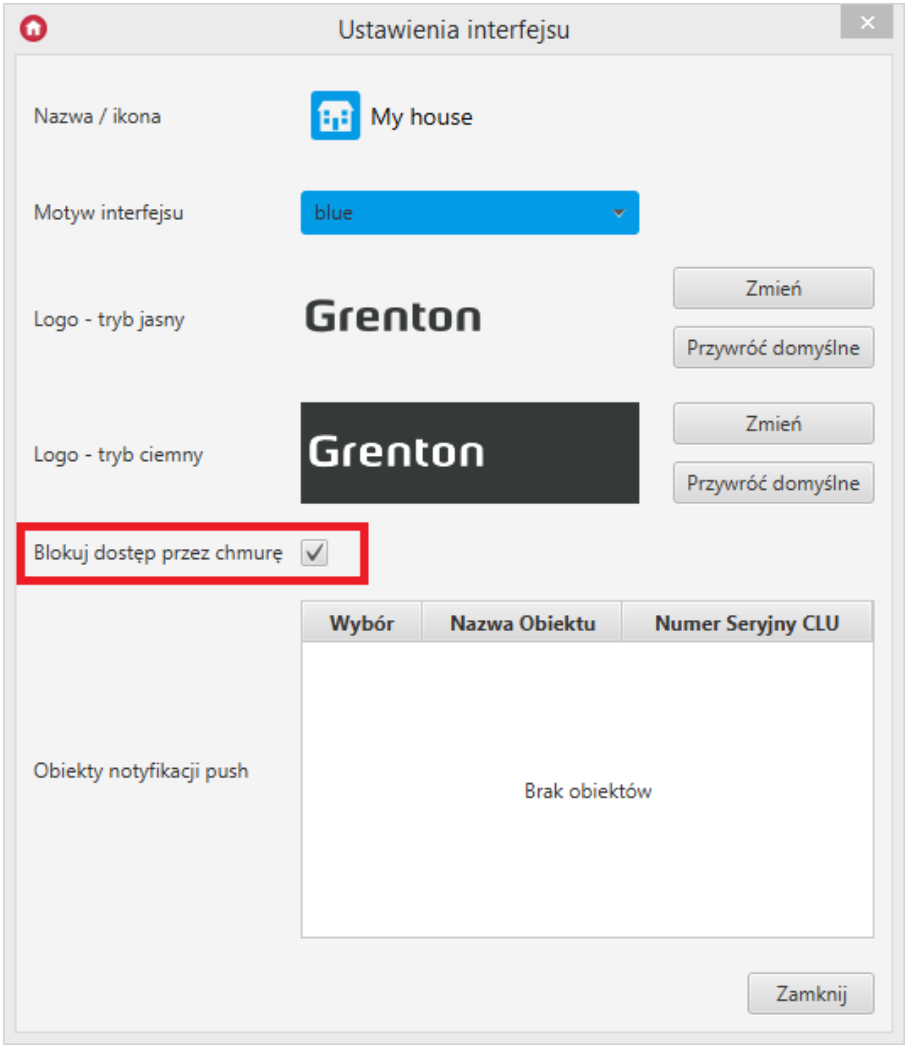
4.3. Zmiana koloru interfejsu

Istnieje również możliwość zmiany koloru motywu interfejsu. Z rozwijanej listy dostępnych jest 15 różnych kolorów motywu.



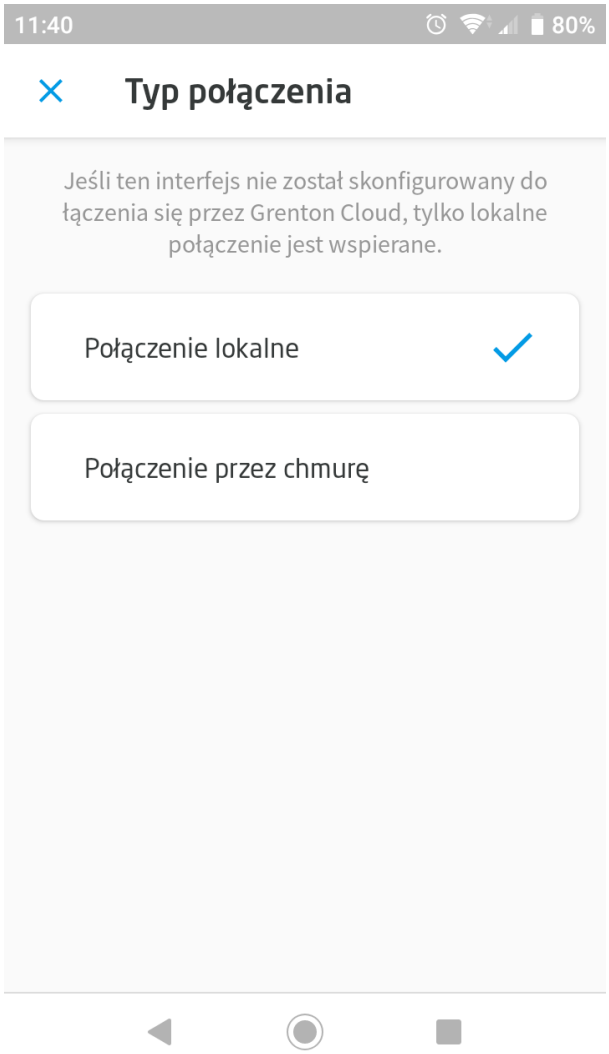
4.4. Blokowanie dostępu przez chmurę

Opcja `Blokuj dostęp przez chmurę` blokuje możliwość połączenia aplikacji z systemem przez chmurę. Możliwe wówczas będzie korzystanie z aplikacji jedynie w trybie połączenia lokalnego.

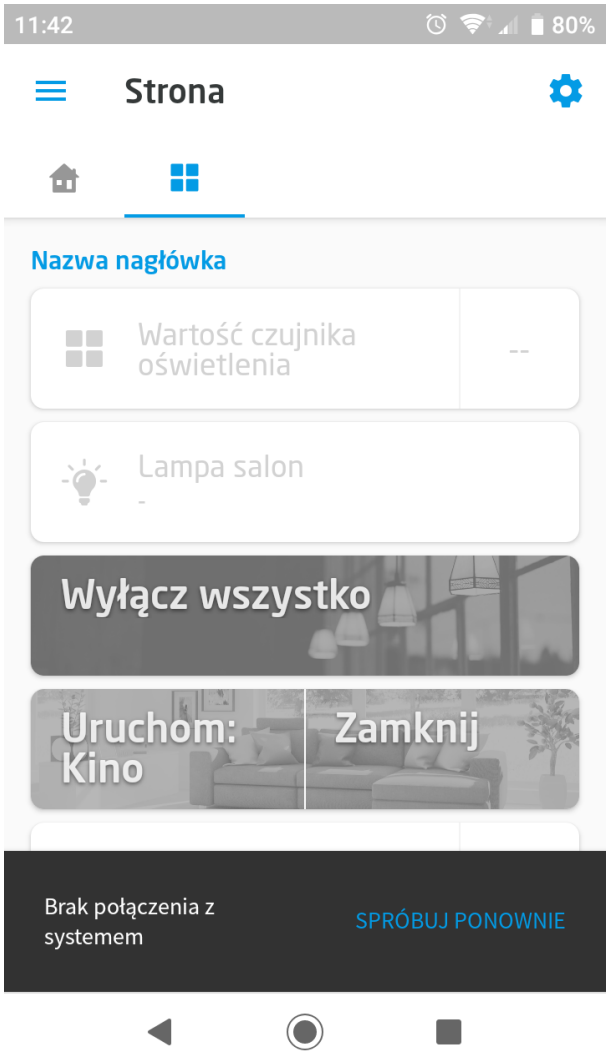


A. Zachowanie aplikacji w systemie Android

Po zablokowaniu dostępu przez chmurę i wysłaniu konfiguracji na urządzenie, tryb połączenia z CLU dla danego interfejsu zostanie ustawiony na połączenie lokalne (w innym wypadku domyślnie ustawiany jest tryb połączenia przez chmurę).

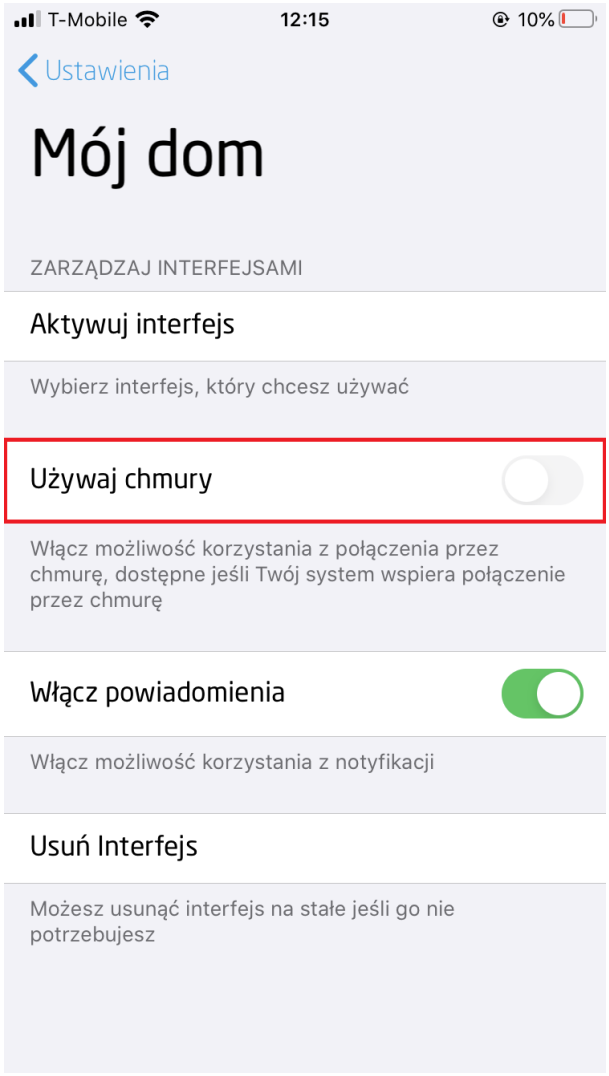


Przy zmianie trybu połączenia na połączenie przez chmurę, połączenie z CLU będzie niemożliwe. Wyświetlony zostanie komunikat o braku połączenia z systemem.



B. Zachowanie aplikacji w systemie iOS

Po zablokowaniu dostępu przez chmurę i wysłaniu konfiguracji na urządzenie, wybranie opcji `Używaj chmury` w danym interfejsie w aplikacji będzie niemożliwe.



5. Wysyłanie interfejsu na urządzenie

Dla utworzonego interfejsu możliwe jest jego wysłanie na urządzenie mobilne na 2 sposoby:

5.1. Wysłanie interfejsu myGrenton poprzez kod QR lub manualnie

Aby wysłać interfejs na telefon należy kliknąć w pasku narzędzi ikonę `Wyślij na telefon interfejs myGrenton`:



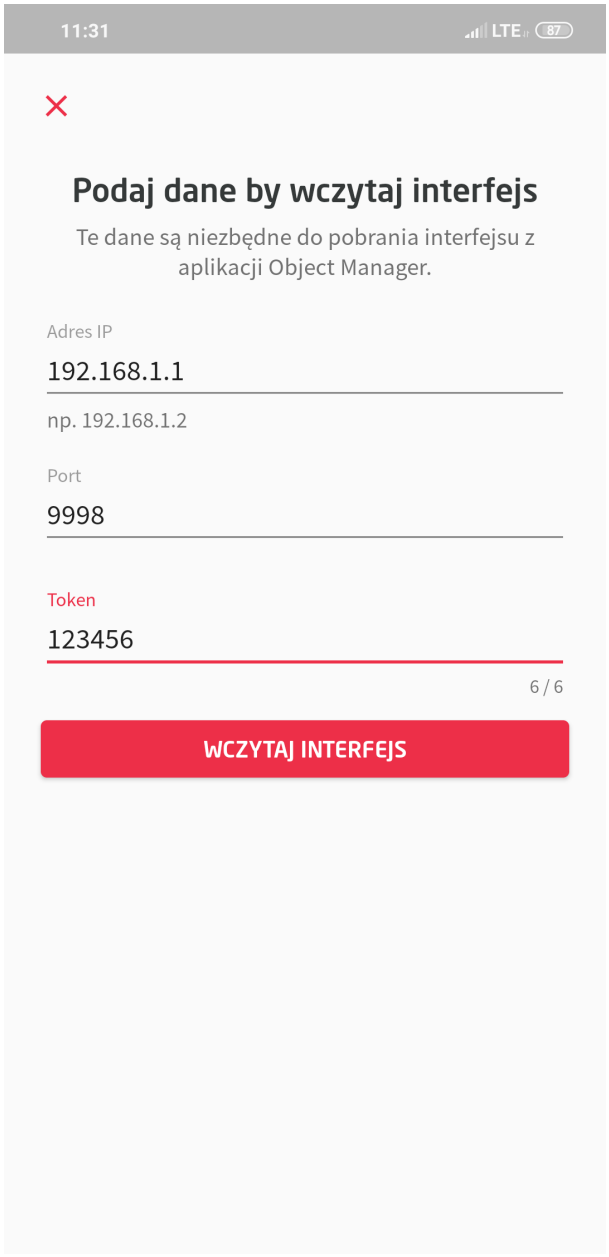
Wyświetlone okno umożliwia przesłanie interfejsu poprzez skanowanie kodu QR bądź podając parametry interfejsu:



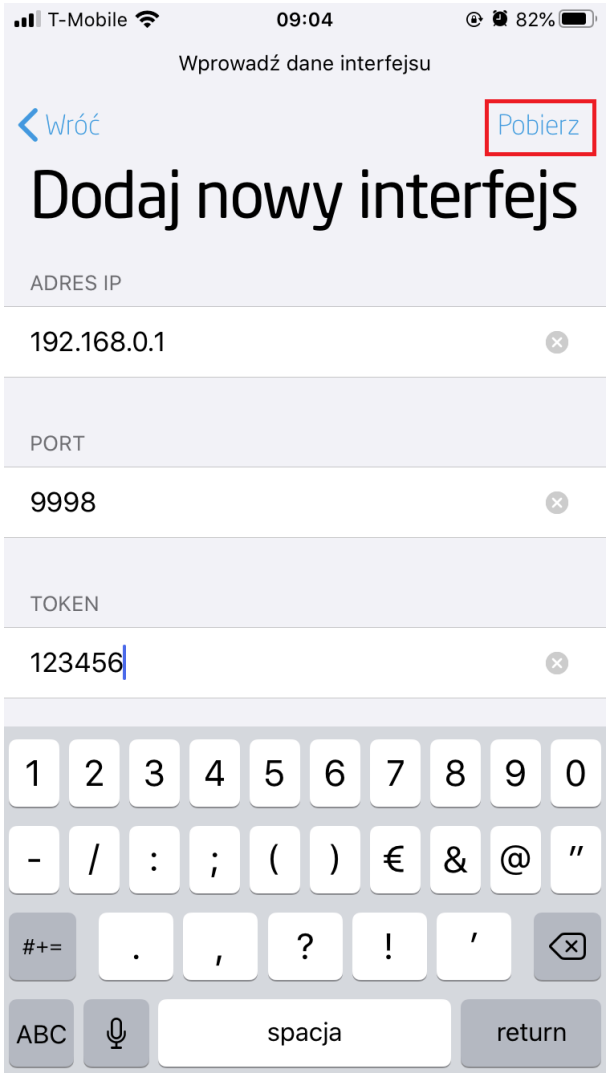
Uwaga! Komputer z aplikacją Object Manager i urządzenie mobilne z aplikacją myGrenton muszą być podłączone do tej samej sieci LAN. Okno dialogowe musi pozostać otwarte do momentu ukończenia przesyłania interfejsu.

Na telefonie należy wybrać opcję skanowania kodu QR lub wpisania ręcznego. W przypadku wpisywania danych, należy uzupełnić pola o dane wyświetlone w oknie Object Managera.

W przypadku aplikacji na urządzenia Android, po podaniu prawidłowych danych, należy wybrać opcję `Wczytaj interfejs`:

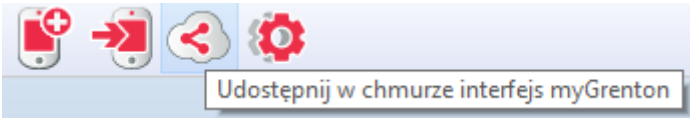


W przypadku iOS, należy wybrać opcję `Pobierz`:

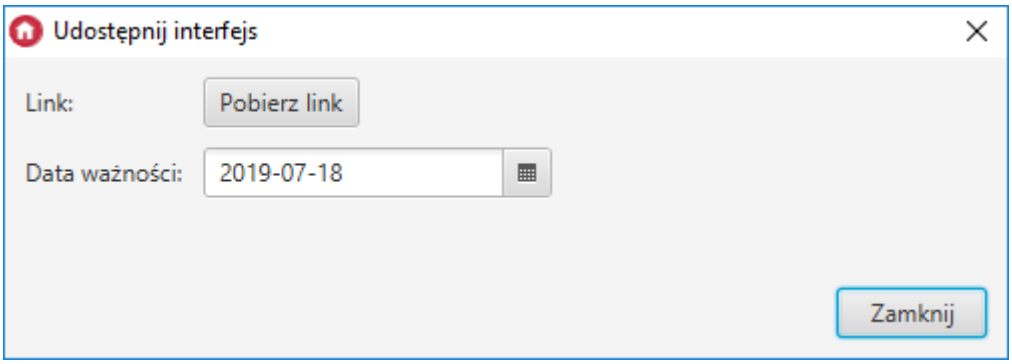


Po prawidłowym przesłaniu aplikacja automatycznie uruchomi wczytany interfejs.

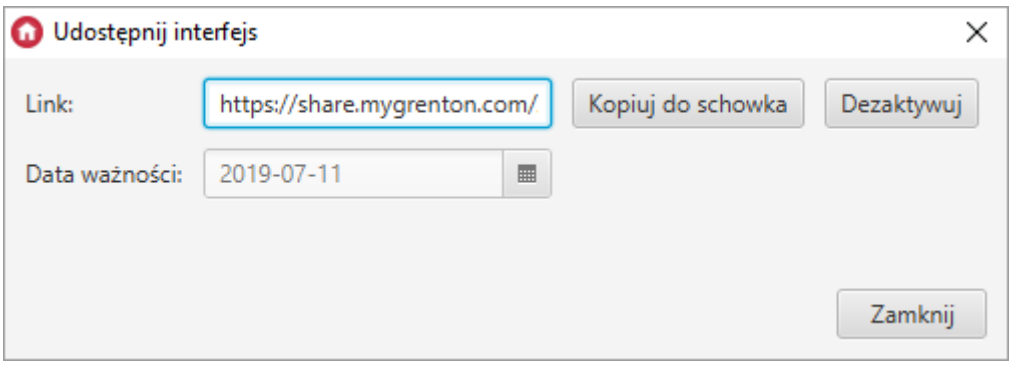
5.2. Udostępnienie interfejsu myGrenton poprzez chmurę



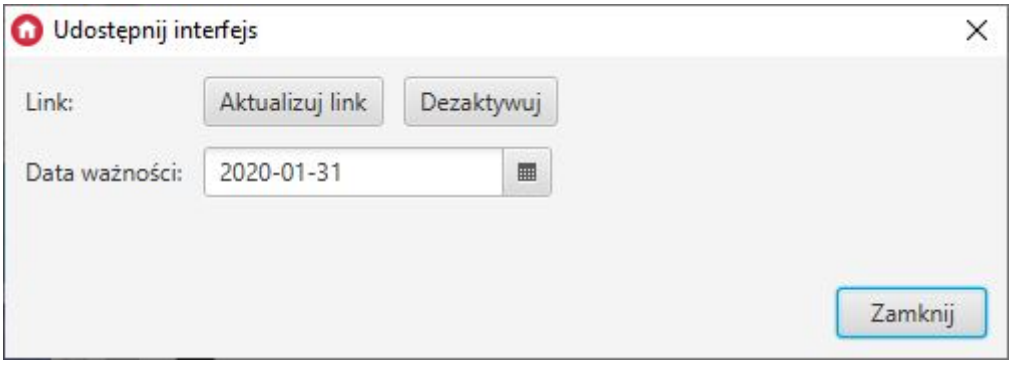
Po kliknięciu ikony `Udostępnij w chmurze interfejs myGrenton` pojawi się okno udostępniania interfejsu do chmury Grenton:



Wygenerowany link można skopiować do schowka. Link można wkleić w dowolną przeglądarkę internetową. Możliwe jest ustawienie daty ważności linku z interfejsem. Po kliknięciu `Pobierz link` pojawia się link do strony z wygenerowanym interfejsem:

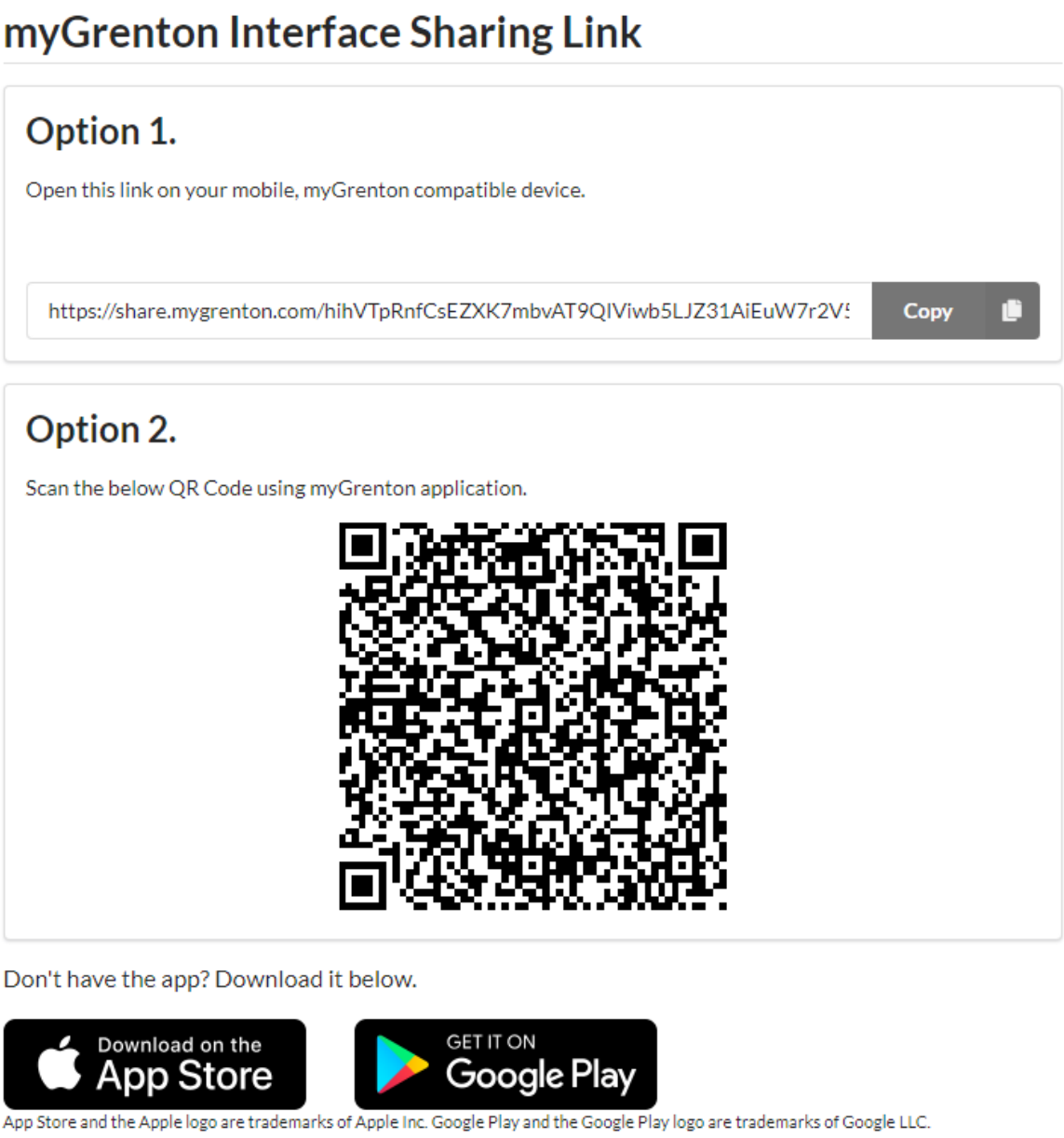


Jeżeli użytkownik wprowadził zmiany w interfejsie, możliwa jest aktualizacja udostępnionego interfejsu:



Po wejściu w udostępniony link pojawi się strona, na której widnieją dwie opcje dodania nowego interfejsu:

1. Poprzez otworzenie wygenerowanego linku w przeglądarce telefonu z systemem Android (nastąpi wówczas automatyczne otworzenie się aplikacji myGrenton oraz dodanie do niej nowego interfejsu) lub poprzez wpisanie linku w aplikacji myGrenton na telefonie z systemem iOS.
2. Poprzez zeskanowanie utworzonego kodu QR w aplikacji myGrenton.



Uwaga! Wysłanie interfejsu, który zawiera widgety nieobsługiwane w danej wersji aplikacji wykona się prawidłowo, widgety nieobsługiwane zostaną pominięte i nie będą widoczne.

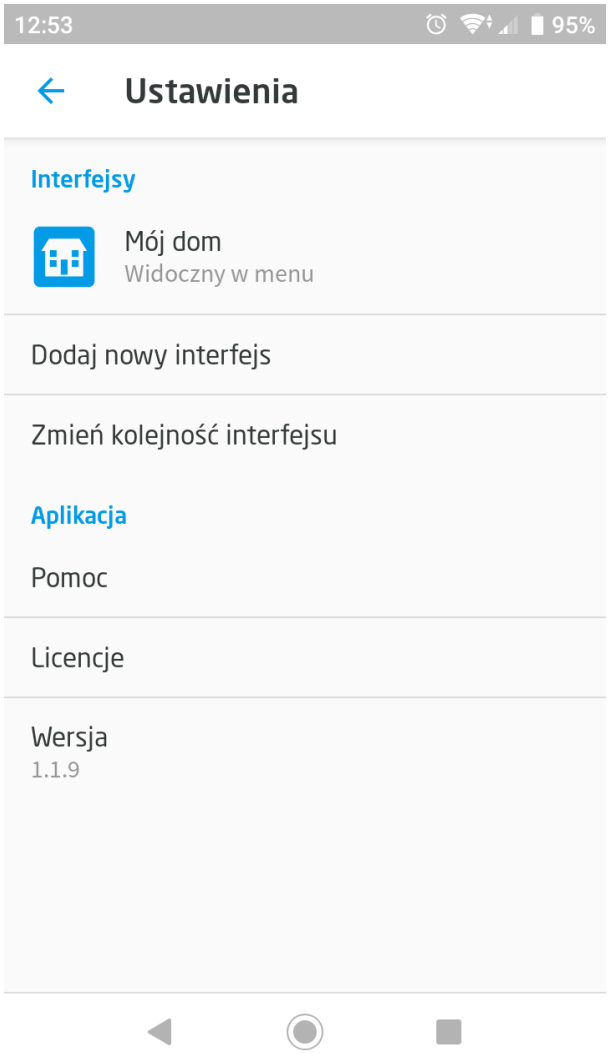
Funkcjonalność dostępna dla aplikacji myGrenton w wersji 1.5.0 (Android) / 1.9.0 (iOS) lub wyższej.

6. Ustawienia aplikacji oraz interfejsu

6.1. Ustawienia aplikacji

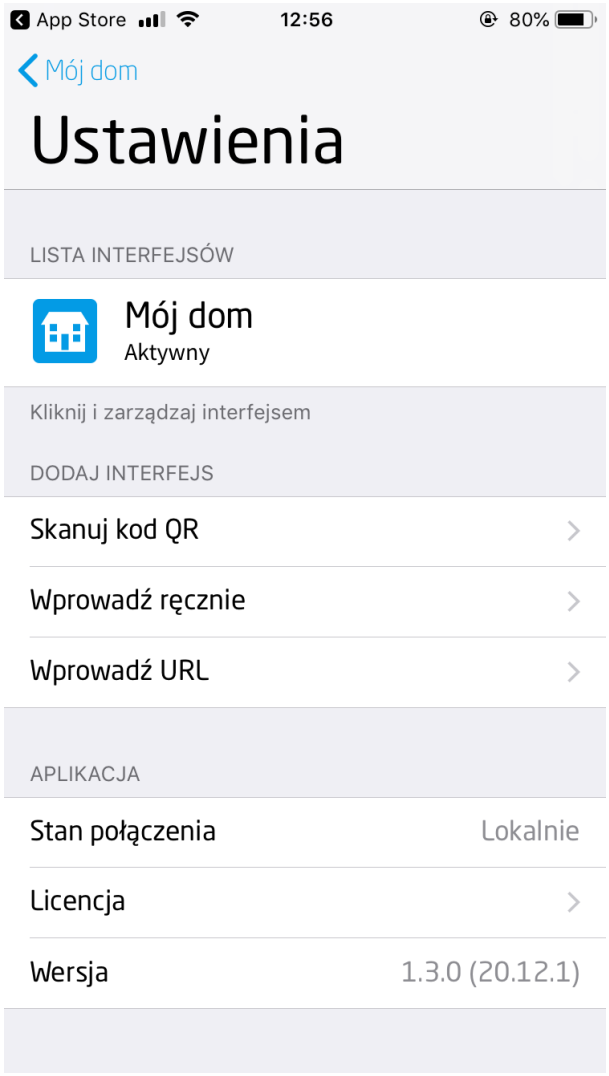
A. Android

W ustawieniach dysponujemy możliwością dodania nowego interfejsu oraz zmiany kolejności interfejsów (dotyczy kolejności wyświetlania interfejsów, pomiędzy którymi możemy się przełączać z poziomu strony głównej aplikacji). Dodatkowo przycisk `Pomoc` przekierowuje na stronę wsparcia technicznego, gdzie można znaleźć przydatne informacje o systemie Grenton. `Licencje` przenoszą użytkownika na nową stronę, gdzie wypisane zostały wszystkie użyte licencje. Pole `Wersja` wyświetla wersję używanej aplikacji.



B. iOS

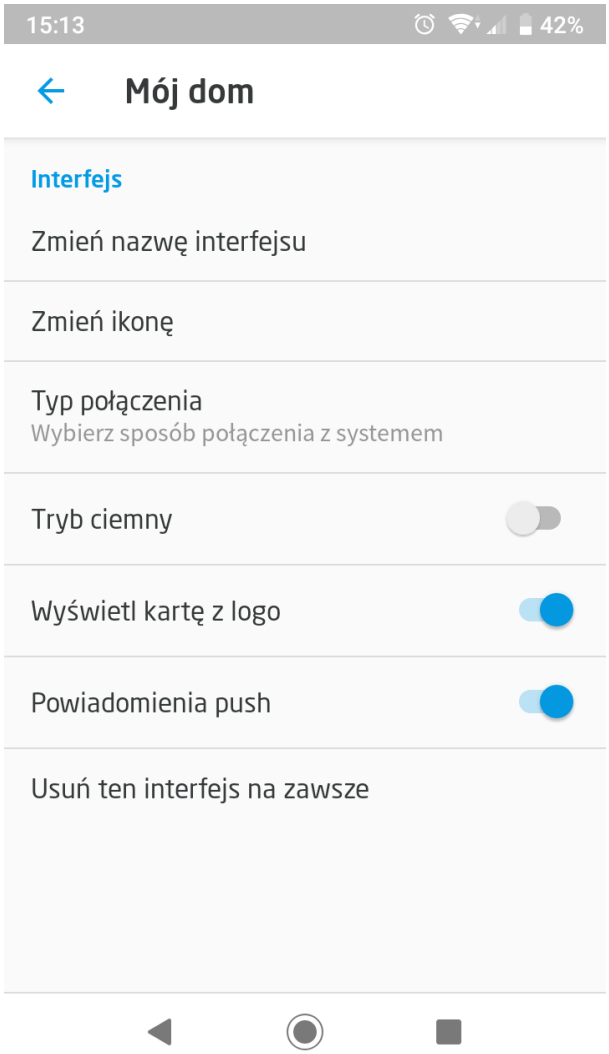
W ustawieniach dysponujemy możliwością dodania nowego interfejsu poprzez skanowanie kodu QR, ręczne wpisanie danych lub wprowadzenie adresu URL interfejsu udostępnionego przez chmurę. Dodatkowo aplikacja pokazuje obecny status połączenia z systemem oraz wersję obecnej aplikacji myGrenton. Przycisk `Licencja` otwiera stronę, gdzie wypisane zostały użyte licencje.



6.2. Ustawienia interfejsu

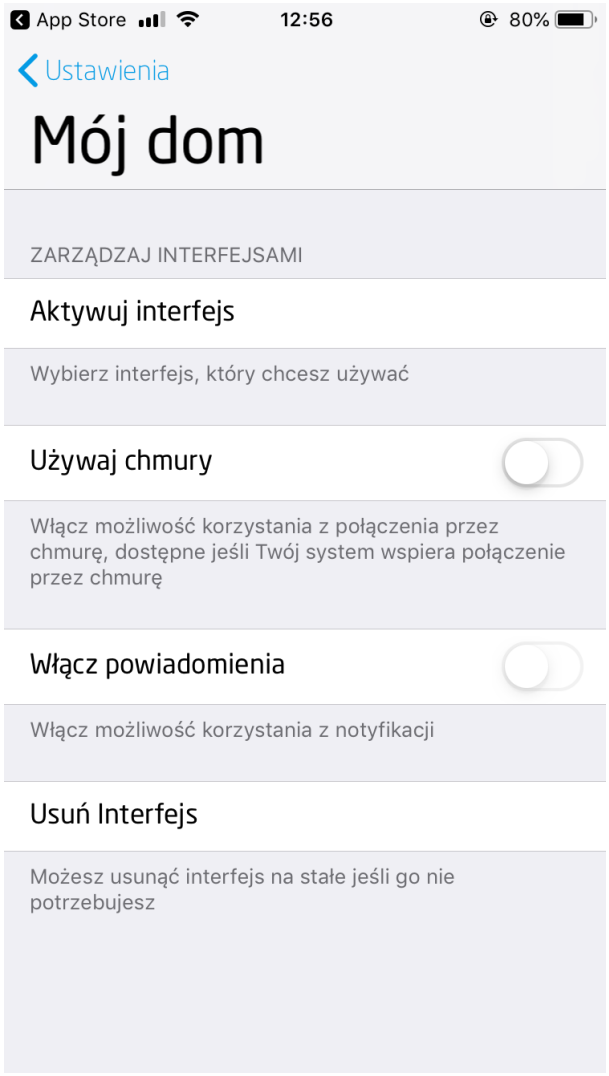
A. Android

Klikając w nazwę interfejsu w ustawieniach przechodzimy do jego ustawień. W ustawieniach interfejsu mamy możliwość zmiany nazwy interfejsu, zmiany ikony, wyboru typu połączenia z systemem (lokalne lub poprzez chmurę), wyboru trybu jasnego bądź ciemnego, wyświetlania karty z logo na stronie głównej, zezwolenia na wyświetlanie notyfikacji push a także usunięcia wybranego interfejsu.



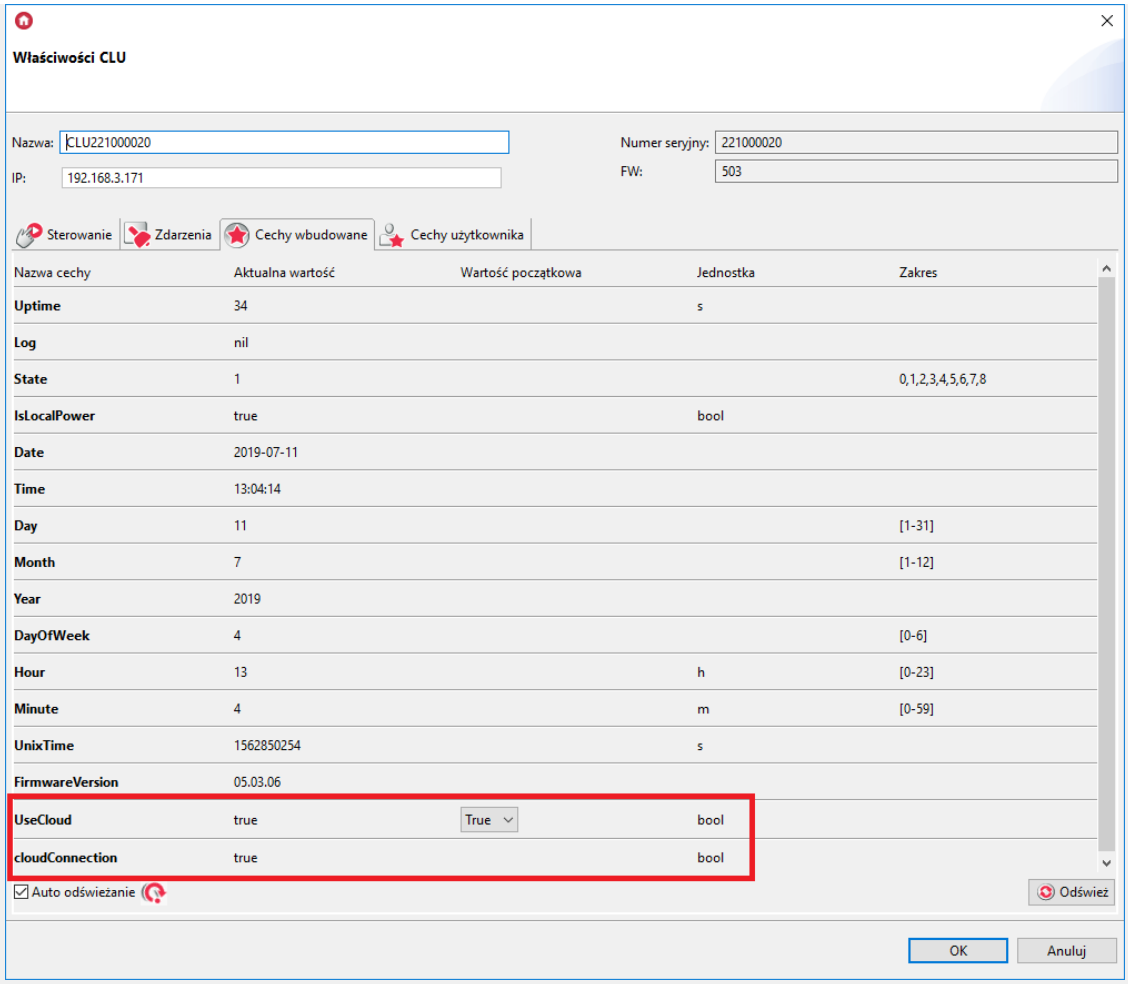
B. iOS

Klikając w dany interfejs w ustawieniach przechodzimy do jego ustawień. W ustawieniach istnieje opcja aktywowania interfejsu. Służy ona do wybrania aktualnie wyświetlanego interfejsu w aplikacji. Dodatkowo możliwy jest wybór czy aplikacja ma korzystać z połączenia przez chmurę - jeśli nie, aplikacja łączy się z systemem jedynie poprzez połączenie lokalne. Ponadto istnieje możliwość włączenia notyfikacji push oraz usunięcia wybranego interfejsu z aplikacji.

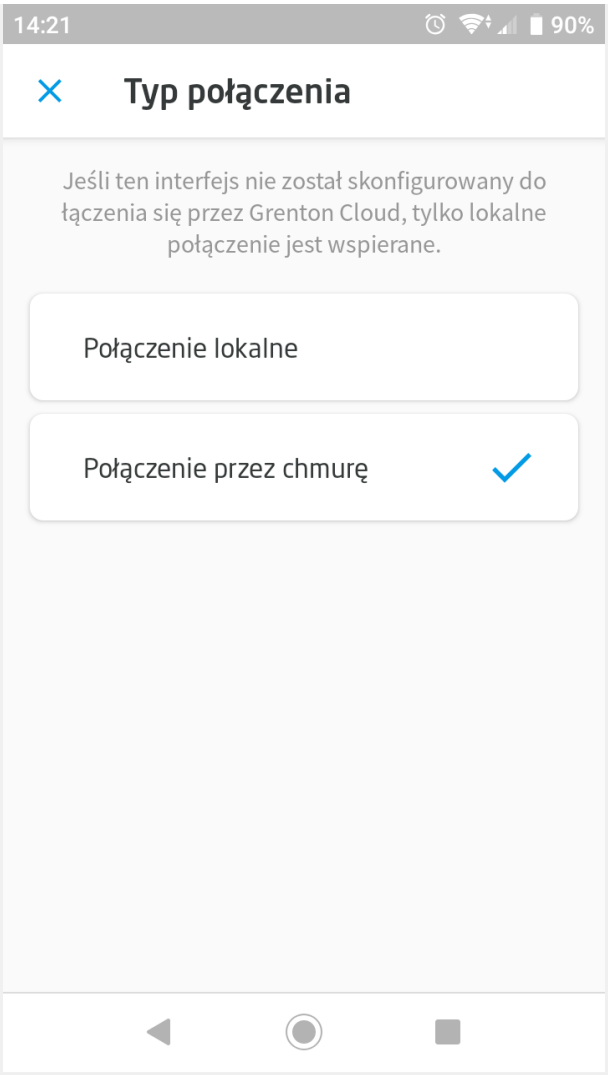


Uwaga! Wymagana minimalna wersja CLU do obsługi chmury: **05.03.06**

Uwaga! Aby poprawnie połączyć się przez chmurę wymagane jest połączenie z chmurą ze strony CLU. Aby to zrobić należy ustawić parametr `UseCloud==true` a następnie wysłać konfigurację na CLU. Poprawne połączenie z chmurą zasygnalizuje parametr `cloudConnection==true`.



Uwaga! Jeżeli CLU połączyło się z chmurą to interfejs wysłany na urządzenie z systemem Android z aplikacją mobilną będzie domyślnie używał połączenia zdalnego. Aby przejść na komunikację lokalną należy wybrać lokalny typ połączenia z systemem. W przypadku urządzeń pracujących na systemie iOS parametr UseCloud jest domyślnie wyłączony.


14:21
90%
Typ połączenia
Jeśli ten interfejs nie został skonfigurowany do łączenia się przez Grenton Cloud, tylko lokalne połączenie jest wspierane.
Połączenie lokalne
Połączenie przez chmurę

XIX. Rozproszenie Logiki Grenton 2.0

> **Uwaga!** Funkcjonalność dostępna jest tylko dla modułów z serii **Grenton 2.0!**

System Grenton w wersji 2.0 posiada funkcjonalność rozproszenia logi i powiązań. Dzięki temu możliwa jest bezpośrednia komunikacja modułów pomiędzy sobą w celu wywoływania akcji między wejściami, a wyjściami, bez pośrednictwa modułu centralnego CLU. W przypadku braku komunikacji modułów wykonawczych z jednostką CLU lub w przypadku awarii CLU, system może dalej funkcjonować w ustalonej konfiguracji. Funkcjonalność jest dostępna dla modułów, które mają załączony tryb Rozproszenia Logiki (Distributed Logic). Docelowo wszystkie urządzenia z serii Grenton 2.0 będą wspierały tą funkcjonalność.

Ustawienie trybu Rozproszenia Logiki dostępny jest dla następujących modułów:

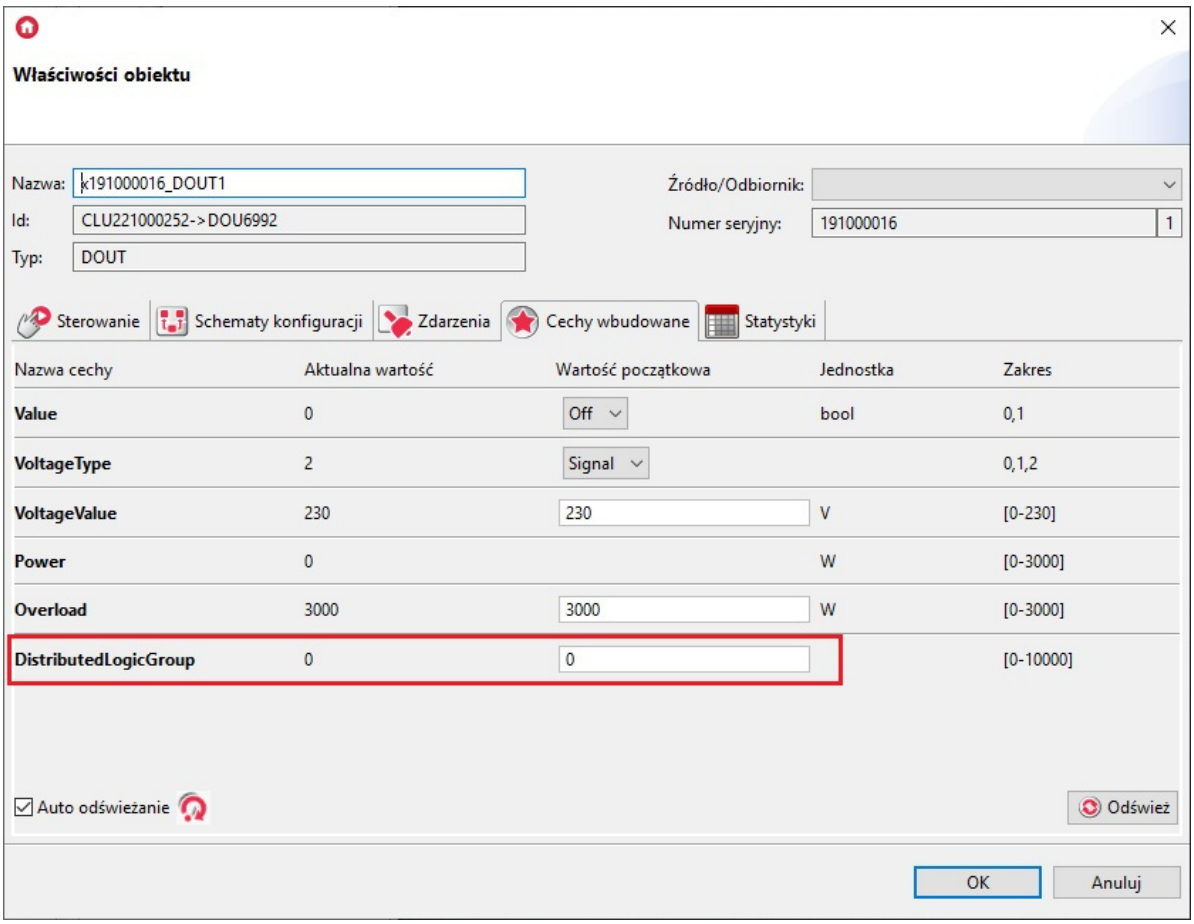
- GRENTON DIGITAL IN 6+3, DIN, TF-Bus (INP-209-D-01)
- GRENTON RELAY 4HP, DIN, TF-Bus (REL-204-D-01)
- GRENTON RELAY 2HP, DIN, TF-Bus (REL-202-D-01)
- GRENTON DIMMER MOSFET, DIN, TF-Bus (DIM-211-D-01)
- GRENTON I/O MODULE 8/8, DIN, TF-Bus (INO-288-D-01)
- GRENTON I/O MODULE 2/2, Flush, TF-Bus, 1-wire (INO-222-T-01)
- GRENTON ROLLER SHUTTER, DIN, TF-Bus (RSH-201-D-01)
- GRENTON ROLLER SHUTTER x3, DIN, TF-Bus (RSH-203-D-01)
- GRENTON ROLLER SHUTTER, Flush, TF-Bus (RSH-201-T-01)
- GRENTON LED RGBW, Flush, TF-Bus (RGB-042-T-16)
- GRENTON TOUCH PANEL 8B, TF-Bus (TPA-208-T-0X)
- GRENTON TOUCH PANEL 4B, TF-Bus (TPA-204-T-0X)
- GRENTON SMART PANEL 4B, OLED, TF-Bus (SPS-204-T-01)

1. Konfiguracja trybu Rozproszenia Logiki

> **Uwaga!** Moduły przy pierwszym podłączeniu do magistrali mają załączony tryb Rozproszenia Logiki - wartość 1 cechy `DistributedLogicGroup` (Default Mode) - wejścia mogą sterować wyjściami, co zostało opisane poniżej. Po wykonaniu CLU Discovery i wysłaniu konfiguracji, następuje wyłączenie trybu.

> **Uwaga!** Moduł Smart Panel przy pierwszym podłączeniu do magistrali ma załączony tryb Rozproszenia Logiki - wartość 1 cech `DistributedLogicGroup_1` - `DistributedLogicGroup_4` (Default Mode) - przypisana jest do czterech fizycznych przycisków - wejścia mogą sterować wyjściami, co zostało opisane poniżej. Po wykonaniu CLU Discovery i wysłaniu konfiguracji, następuje wyłączenie trybu.

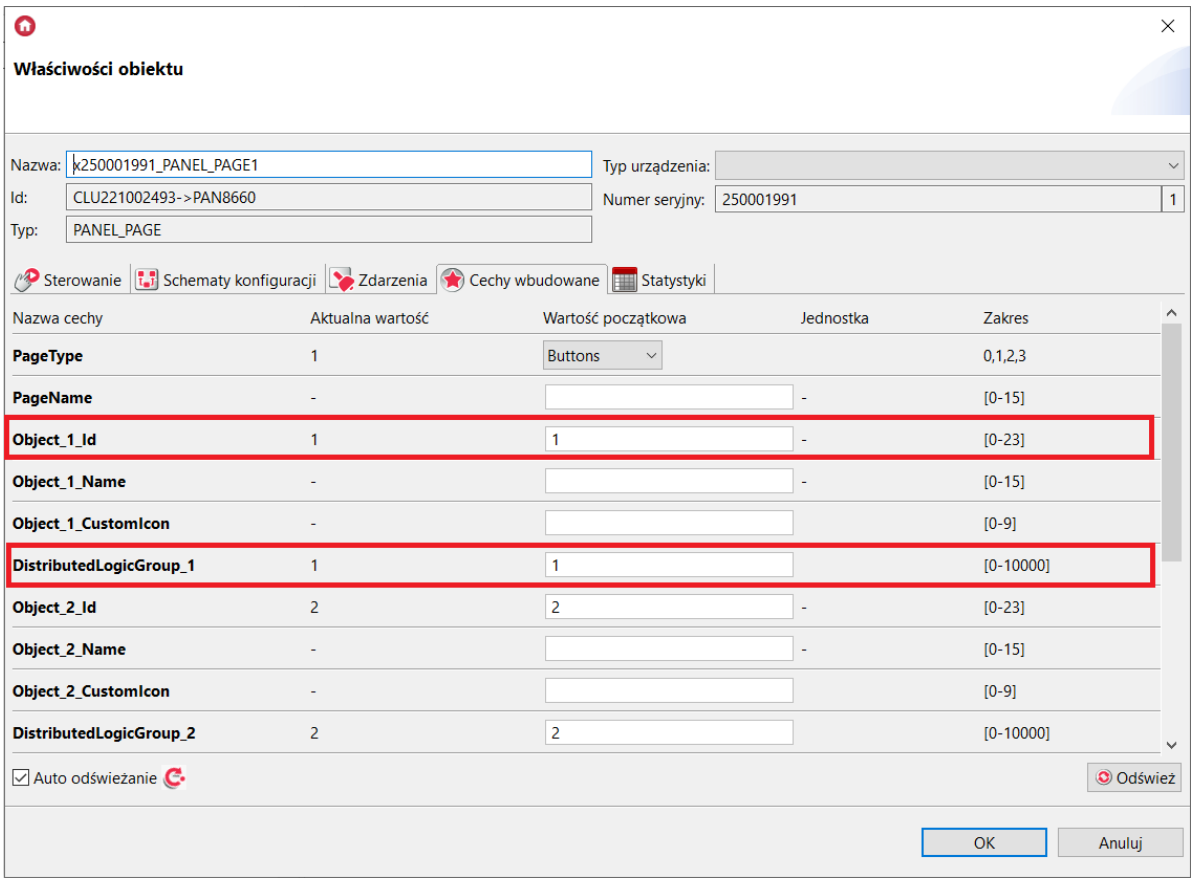
A. Konfiguracja trybu dla modułów IN/OUT



Konfiguracja Logiki Rozproszonej odbywa się tak samo jak dla każdej innej funkcjonalności systemu za pośrednictwem aplikacji OM i definiowana jest dla każdego obiektu IN/OUT danego modułu. Służy do tego cecha `DistributedLogicGroup`. Domyślna wartość `DistributedLogicGroup` jest równa 0, co oznacza wyłączenie trybu.

W przypadku utraty połączenia modułów z CLU oraz wykrycia zdarzenia dla obiektu wejściowego (obiekt IN danego modułu) rozsyłana jest wiadomość do każdego obiektu wyjściowego (obiekt OUT danego modułu), który posiadają tą samą wartość cechy `DistributedLogicGroup`. W wyniku otrzymania wiadomości na obiekcie wyjściowym zostaje wywołana odpowiednia akcja przypisana do otrzymanego zdarzenia.

B. Konfiguracja trybu dla modułu Smart Panel



W przypadku modułu Smart Panel powiązanie po stronie wejścia odbywa się w obiekcie `PANEL_PAGEx`. Wcześniej należy powiązać obiekty `PANEL_PAGEx` z odpowiednimi obiektami `PANEL_BUTTONx` poprzez ustawienie odpowiednich wartości cech `Object_x_Id`. W przypadku braku połączenia z CLU, w ustawionym trybie Logiki Rozproszonej mogą działać wszystkie 16 przycisków wraz ze zmianą stron za pomocą gestów.

> **Uwaga!** Działanie Logiki Rozproszonej możliwe jest tylko w trybie `Buttons` obiektu `PANEL_PAGEx`.

1.1. Działanie Logiki Rozproszonej pomiędzy obiektami DIN oraz wyjściowymi

> **Uwaga!** Zdarzenia oraz wywoływane akcje są ustawione statycznie i nie ma możliwości ich zmiany.

Dostępne akcje podczas działania w trybie Rozproszenia Logiki:

A. Obiekt DIN oraz DOUT

- Załączenie obiektu wejściowego DIN (SwitchOn) --> Załączenie danego wyjścia DOUT.
- Wyłączenie obiektu wejściowego DIN (SwitchOff) --> Wyłączenie danego wyjścia DOUT.
- Krótka zmiana stanu obiektu wejściowego DIN (Click) --> Zmiana stanu na przeciwny danego wyjścia DOUT.

B. Obiekt DIN oraz DIMM

- Załączenie obiektu wejściowego DIN (SwitchOn) --> Załączenie danego wyjścia DIMM .
- Wyłączenie obiektu wejściowego DIN (SwitchOff) --> Wyłączenie danego wyjścia DIMM.
- Krótka zmiana stanu obiektu wejściowego DIN (Click) --> Zmiana stanu na przeciwny danego wyjścia DIMM.

C. Obiekt DIN oraz LEDRGBW

- Załączenie obiektu wejściowego DIN (SwitchOn) --> Załączenie danego kanału LEDRGBW.
- Wyłączenie obiektu wejściowego DIN (SwitchOff) --> Wyłączenie danego kanału LEDRGBW.
- Krótka zmiana stanu obiektu wejściowego DIN (Click) --> Zmiana stanu na przeciwny danego kanału LED RGBW.

D. Obiekt DIN oraz ROLLER_SHUTTER

- Załączenie obiektu wejściowego DIN (SwitchOn) --> Załączenie przekaźnika UP lub DOWN ROLLER_SHUTTER w zależności od poprzedniego kierunku jazdy.
- Wyłączenie obiektu wejściowego DIN (SwitchOff) --> Wyłączenie danego załączonego przekaźnika (UP lub DOWN).
- Krótka zmiana stanu obiektu wejściowego DIN (Click) --> Zmiana stanu na przeciwny ROLLER_SHUTTER:
    - jeśli przekaźniki są wyłączone - załączenie przekaźnika UP lub DOWN, w zależności od poprzedniego kierunku jazdy,
    - jeśli przekaźnik UP lub DOWN jest załączony - wyłączenie przekaźnika.

> **Uwaga!** Załączenie przekaźnika UP lub DOWN jest załączeniem przekaźnika bez wyłączenia po upłynięciu czasu MaxTime. Przekaźniki należy wyłączyć danym wejściem sterującym obiektem ROLLER_SHUTTER w trybie Rozproszonej Logiki.

1.2. Działanie Logiki Rozproszonej pomiędzy obiektami BUTTON oraz wyjściowymi

> **Uwaga!** Zdarzenia oraz wywoływane akcje są ustawione statycznie i nie ma możliwości ich zmiany.

Dostępne akcje podczas działania w trybie Rozproszenia Logiki:

A. Obiekt BUTTON oraz DOUT

- Naciśnięcie i przytrzymanie przycisku (SwitchOn) --> Załączenie danego wyjścia DOUT.
- Zwolnienie przycisku (SwitchOff) --> Wyłączenie danego wyjścia DOUT.
- Krótkie naciśnięcie przycisku (Click) --> Zmiana stanu na przeciwny DOUT.

B. Obiekt BUTTON oraz DIMM

- Naciśnięcie i przytrzymanie przycisku (SwitchOn) --> Załączenie danego wyjścia DIMM .
- Zwolnienie przycisku (SwitchOff) --> Wyłączenie danego wyjścia DIMM .
- Krótkie naciśnięcie przycisku (Click) --> Zmiana stanu na przeciwny danego wyjścia DIMM.

C. Obiekt BUTTON oraz LEDRGBW

- Naciśnięcie i przytrzymanie przycisku (SwitchOn) --> Załączenie danego kanału LEDRGBW.
- Zwolnienie przycisku (SwitchOff) --> Wyłączenie danego kanału LEDRGBW.
- Krótkie naciśnięcie przycisku (Click) --> Zmiana stanu na przeciwny danego kanału LEDRGBW.

D. Obiekt BUTTON oraz ROLLER_SHUTTER

- Naciśnięcie i przytrzymanie przycisku (SwitchOn) --> Załączenie przekaźnika UP lub DOWN ROLLER_SHUTTER w zależności od poprzedniego kierunku jazdy.
- Zwolnienie przycisku (SwitchOff) --> Wyłączenie danego załączonego przekaźnika (UP lub DOWN).
- Krótkie naciśnięcie przycisku (Click) --> Zmiana stanu na przeciwny ROLLER_SHUTTER:
    - jeśli przekaźniki są wyłączone - załączenie przekaźnika UP lub DOWN, w zależności od poprzedniego kierunku jazdy,
    - jeśli przekaźnik UP lub DOWN jest załączony - wyłączenie przekaźnika.

> **Uwaga!** Załączenie przekaźnika UP lub DOWN jest załączeniem przekaźnika bez wyłączenia po upłynięciu czasu MaxTime. Przekaźniki należy wyłączyć danym wejściem sterującym obiektem ROLLER_SHUTTER w trybie Rozproszonej Logiki.

1.3. Działanie Logiki Rozproszonej pomiędzy obiektami PANEL_PAGE z przypisanymi PANEL_BUTTON oraz wyjściowymi

> **Uwaga!** Zdarzenia oraz wywoływane akcje są ustawione statycznie i nie ma możliwości ich zmiany.

Dostępne akcje podczas działania w trybie Rozproszenia Logiki:

A. Obiekt PANEL_PAGE z ustawionym obiektem PANEL_BUTTON oraz DOUT

- Naciśnięcie i przytrzymanie przycisku (SwitchOn) --> Załączenie danego wyjścia DOUT.
- Zwolnienie przycisku (SwitchOff) --> Wyłączenie danego wyjścia DOUT.
- Krótkie naciśnięcie przycisku (Click) --> Zmiana stanu na przeciwny DOUT.

B. Obiekt PANEL_PAGE z ustawionym obiektem PANEL_BUTTON oraz DIMM

- Naciśnięcie i przytrzymanie przycisku (SwitchOn) --> Załączenie danego wyjścia DIMM .
- Zwolnienie przycisku (SwitchOff) --> Wyłączenie danego wyjścia DIMM .
- Krótkie naciśnięcie przycisku (Click) --> Zmiana stanu na przeciwny danego wyjścia DIMM.

C. Obiekt PANEL_PAGE z ustawionym obiektem PANEL_BUTTON oraz LEDRGBW

- Naciśnięcie i przytrzymanie przycisku (SwitchOn) --> Załączenie danego kanału LEDRGBW.
- Zwolnienie przycisku (SwitchOff) --> Wyłączenie danego kanału LEDRGBW.
- Krótkie naciśnięcie przycisku (Click) --> Zmiana stanu na przeciwny danego kanału LEDRGBW.

D. Obiekt PANEL_PAGE z ustawionym obiektem PANEL_BUTTON oraz ROLLER_SHUTTER

- Naciśnięcie i przytrzymanie przycisku (SwitchOn) --> Załączenie przekaźnika UP lub DOWN ROLLER_SHUTTER w zależności od poprzedniego kierunku jazdy.
- Zwolnienie przycisku (SwitchOff) --> Wyłączenie danego załączonego przekaźnika (UP lub DOWN).
- Krótkie naciśnięcie przycisku (Click) --> Zmiana stanu na przeciwny ROLLER_SHUTTER:
    - jeśli przekaźniki są wyłączone - załączenie przekaźnika UP lub DOWN, w zależności od poprzedniego kierunku jazdy,
    - jeśli przekaźnik UP lub DOWN jest załączony - wyłączenie przekaźnika.

> **Uwaga!** Załączenie przekaźnika UP lub DOWN jest załączeniem przekaźnika bez wyłączenia po upłynięciu czasu MaxTime. Przekaźniki należy wyłączyć danym wejściem sterującym obiektem ROLLER_SHUTTER w trybie Rozproszonej Logiki.

2. Default Mode

W przypadku, gdy do danego obiektu wartość cechy `DistributedLogicGroup` ustawiona jest na wartość 1, działa on w trybie **Default Mode**. Jest to specjalny tryb działania, domyślnie ustawiany dla każdego obiektu.

> **Uwaga!** Zdarzenia oraz wywoływane akcje są ustawione statycznie i nie ma możliwości ich zmiany.

2.1. Default Mode dla modułów wejściowych oraz modułów wyjściowych

- moduł wejściowy (DIGITAL IN 6+3 DIN) - steruje wszystkimi modułami wyjściowymi (RELAY 4HP DIN, RELAY 2HP DIN, DIMMER MOSFET DIN, ROLLER SHUTTER DIN, ROLLER SHUTTER DIN) w sieci TFBUS będącymi również w Default Mode, przykładowo:
    - DIGITAL IN1 --> RELAY 4HP OUT1 | RELAY 2HP OUT1 | DIMMER MOSFET DIMM1 | ROLLER_SHUTTER1.
    - DIGITAL IN2 --> RELAY 4HP OUT2 | RELAY 2HP OUT2 | ROLLER_SHUTTER2.
    - DIGITAL IN3 --> RELAY 4HP OUT3 | ROLLER_SHUTTER3.
    - DIGITAL IN4 --> RELAY 4HP OUT4.
- moduł Touch Panel (TOUCH PANEL 4B, TOUCH PANEL 8B) - steruje wszystkimi modułami wyjściowymi (RELAY 4HP DIN, RELAY 2HP DIN, DIMMER MOSFET DIN, ROLLER SHUTTER DIN) w sieci TFBUS będącymi również w Default Mode, przykładowo:
    - TOUCH PANEL BUTTON1 --> RELAY 4HP OUT1 | RELAY 2HP OUT1 | DIMMER MOSFET DIMM1 | ROLLER_SHUTTER1.
    - TOUCH PANEL BUTTON2 --> RELAY 4HP OUT2 | RELAY 2HP OUT2 | ROLLER_SHUTTER2.
    - TOUCH PANEL BUTTON3 --> RELAY 4HP OUT3 | ROLLER_SHUTTER3.
    - TOUCH PANEL BUTTON4 --> RELAY 4HP OUT4.
- moduł Smart Panel - steruje wszystkimi modułami wyjściowymi (RELAY 4HP DIN, RELAY 2HP DIN, DIMMER MOSFET DIN, ROLLER SHUTTER DIN) w sieci TFBUS będącymi również w Default Mode, przykładowo:

- SMART PANEL PANEL_PAGE1 cecha DistributedLogicGroup_1 --> RELAY 4HP OUT1 | RELAY 2HP OUT1 | DIMMER MOSFET DIMM1 | ROLLER_SHUTTER1.
- SMART PANEL PANEL_PAGE1 cecha DistributedLogicGroup_2 --> RELAY 4HP OUT2 | RELAY 2HP OUT2 | ROLLER_SHUTTER2.
- SMART PANEL PANEL_PAGE1 cecha DistributedLogicGroup_3 --> RELAY 4HP OUT3 | ROLLER_SHUTTER3.
- SMART PANEL PANEL_PAGE1 cecha DistributedLogicGroup_4 --> RELAY 4HP OUT4.

Uwaga! Załączenie przekaźnika UP lub DOWN jest załączeniem przekaźnika bez wyłączenia po upłynięciu czasu MaxTime. Przekaźniki należy wyłączyć danym wejściem sterującym Roller Shutterem w trybie Rozproszonej Logiki.

Uwaga!

Dla modułu Smart Panel działającego w Default Mode Logiki Rozproszonej zachowanie opisane powyżej jest identyczne dla każdego obiektu PANEL_PAGE (PANEL_PAGE1 - PANEL_PAGE4).

2.2. Default Mode dla modułów posiadających własne wejścia / wyjścia

- moduł posiadający wejścia oraz wyjścia (I/O MODULE DIN 8, I/O MODULE FM, ROLLER SHUTTER FM) - steruje własnymi kanałami o adekwatnym numerze (IN1->OUT1, IN2->OUT2, itd), przykładowo:
  - I/O MODULE FM IN1 --> I/O MODULE FM OUT1.
  - I/O MODULE FM IN2 --> I/O MODULE FM OUT2.
  - ROLLER SHUTTER FM IN1 --> ROLLER_SHUTTER1.
- moduł LED RGBW FM - steruje własnymi kanałami (Red, Green):
  - LED RGBW FM IN1 --> LED RGBW FM kanał R.
  - LED RGBW FM IN2 --> LED RGBW FM kanał G.

3. Przywrócenie komunikacji pomiędzy CLU a modułem

W momencie, gdy komunikacja pomiędzy CLU a modułami zostanie przywrócona, wartość cech Value danych obiektów zostaje zaktualizowana do rzeczywistej wartości wejść / wyjść (na wartość zmienioną podczas działania trybu Rozproszenia Logiki), a moduły wykonują działania zgodnie z zaprogramowaną logiką na CLU.

---

1. W zależności od rodzaju używanego routera, jego interfejs może się różnić od ogólnej instrukcji konfiguracji portów. ↩
2. Jest to domyślny port dla potrzeb obsługi strumienia kamery *rtsp*. ↩
3. Jego adres IP można znaleźć na liście aktualnie podłączonych do sieci urządzeń w interfejsie routera. ↩
4. W zależności od tego jakiego rodzaju urządzenie jest w użyciu, jego konfiguracja może się różnić od podanej w instrukcji. ↩
5. Oprócz ustawień połączenia w tej samej sekcji istnieje możliwość zaznaczenia pola wyboru, decydującego o używaniu trybu głośnomówiącego po odebraniu połączenia. ↩
6. Gdzie X i Y oznaczają nazwy CLU. ↩
7. W rozumieniu instrukcji słowo składa się z dwóch bajtów. ↩