

# DOMIQ™



## DOMIQ/Base

# Instrukcja konfiguracji

CM-BL-PL-250129

**Copyright 2025 DOMIQ Sp. z o.o.**

Wszystkie prawa zastrzeżone.

Styczeń 2025.

*DOMIQ oraz logo są zastrzeżonymi znakami towarowymi DOMIQ Sp. z o.o.*

*LCN jest zastrzeżony znakiem towarowym Issendorff KG.*

*SATEL jest zastrzeżonym znakiem towarowym SATEL Sp. z o.o.*

*iPhone, iPod Touch i iPad są zastrzeżonymi znakami towarowymi Apple Inc.*

Dokumentacja opisuje produkty i oprogramowanie DOMIQ w najnowszej wersji dostępnej w chwili opublikowania niniejszej dokumentacji. W związku z ciągłym rozwojem produktów możliwe są drobne zmiany w wyglądzie i funkcjach poszczególnych ekranów. W przypadku wątpliwości prosimy o kontakt:

Skype: **domiq-support**

Email: **info@domiq.pl**

Aktualizacje dokumentacji są dostępne na stronach internetowych **www.domiq.pl**.

# Spis treści

2.1. DOMIQ/Base .....	12
2.2. DOMIQ/Serial .....	13
2.3. DOMIQ/Light .....	13
2.4. DOMIQ/Expander .....	13
2.5. DOMIQ/Remote.....	14
3.1. Podłączenie.....	15
3.2. Pierwsze logowanie .....	16
1. Wyszukiwanie adresu IP modułu DOMIQ/Base.....	16
2. Uruchomienie konfiguratora .....	16
3.3. Ekran startowy .....	17
3.4. Ustawienie danych dostępowych .....	17
1. Dostęp dla użytkownika.....	17
2. Dostęp administracyjny.....	17
3. Programowanie LCN .....	18
4.1. Ustawienia ogólne .....	20
4.2. Ustawienia sieciowe .....	21
4.3. Parowanie aplikacji .....	22
4.3.1. Autoryzacje dla aplikacji .....	22
4.4. Integracje.....	24
4.4.1. Satel .....	24
4.4.2. BACnet.....	24
4.4.3. IFTTT .....	25
4.4.4. UPnP .....	25
4.4.5. u::Lux .....	25
4.4.6. MQTT.....	25
4.4.7. Połączenia szeregowo.....	25
5.0.1. Stan.....	28
5.0.2. Zmienne MEM.....	29
6.0.1. Statystyki pamięci .....	31
6.0.2. Struktura plików.....	31
6.0.3. Okno podglądu .....	33
7.1. Remote .....	36
7.1.1. Struktura .....	37
1. Strona .....	37
2. Sekcja.....	37
7.1.2. Kontrolki kanałowe .....	38
1. Grupa przycisków .....	38
2. Przycisk .....	38
3. Kamera.....	38
4. Wykres.....	39
5. Przełącznik.....	39
6. Ściemniacz.....	39
7. Światło RGBW .....	40
8. Wartość .....	40
9. Roleta.....	41
10. Status .....	41
11. Regulator .....	42
12. Tekst.....	42
13. Wybór.....	42
14. Czas.....	44

7.1.3. LCN.....	45
1. Ściemniacz.....	45
2. Przełącznik.....	45
3. Temperatura.....	46
4. Roleta.....	46
5. Wejście binarne LCN.....	47
7.1.4. IDS.....	48
1. Wejście IDS.....	48
2. Wyjście IDS.....	48
3. Strefa IDS.....	48
7.2. Wizualizacja.....	49
7.2.1. Pasek przycisków.....	49
7.2.2. Tabela atrybutów.....	49
7.2.3. Struktura.....	49
1. Ekran.....	50
2. Warstwa.....	51
7.2.4. Edytor wizualizacji.....	53
1. Światło.....	54
2. Przycisk.....	56
3. Przełącznik.....	56
4. Ściemniacz.....	57
5. Wartość.....	57
6. Temperatura.....	58
7. Tekst.....	58
8. RGBW.....	58
9. Kamera.....	59
10. Obrazek.....	59
8.1. Akcje.....	62
8.1.1. + LCN.....	62
1. Wyjście.....	62
2. Przekazniki.....	62
3. Silniki.....	62
4. Wyślij klawisze.....	63
5. Zablokuj klawisze.....	63
6. Scena.....	63
7. Rejestr.....	63
8.1.2. + VAR.....	63
8.1.3. + Komenda.....	63
8.2. Zdarzenia.....	65
8.2.1. Pasek przycisków.....	65
8.2.2. Struktura.....	65
1. Grupa.....	66
2. Zdarzenie.....	66
8.2.3. Szczegóły.....	66
8.2.4. Akcje.....	69
8.2.5. Zagadnienia zaawansowane.....	70
8.3. Timery.....	75
8.3.1. Szczegóły.....	75
8.3.2. Zegar astronomiczny.....	76
8.3.3. Timery wywoływane komendami.....	76
1. TIMER.....	76
2. DELAY.....	79
8.3.4. Zagadnienia zaawansowane.....	80
8.4. Logika.....	83
8.4.1. Uwagi ogólne.....	83
8.4.2. Moduły i funkcje standardowe.....	84
8.4.3. Moduły i funkcje niestandardowe.....	84
1. Moduły.....	84

2. Funkcje.....	95
9.1. Ekspandery.....	100
9.1.1. Główny.....	100
1. Tabela wejść.....	100
2. Tabela wyjść.....	101
9.1.2. Grupy.....	101
1. Tabela grup.....	101
9.1.3. Wyjścia.....	102
9.1.4. Sterowanie.....	102
9.2. MODBUS.....	103
9.2.1. Etapy integracji.....	103
9.2.2. Interfejs konfiguracyjny.....	104
9.2.3. Dodawanie połączenia.....	104
9.2.4. Dodawanie urządzenia.....	105
9.2.5. Konfiguracja rejestrów.....	105
9.2.6. Usuwanie problemów.....	106
9.3. DALI.....	108
9.3.1. Etapy integracji.....	108
9.3.2. Interfejs konfiguracyjny.....	109
1. Tabela balastów.....	109
2. Wszystkie balasty.....	109
3. Pojedynczy balast.....	110
9.3.3. Sterowanie z LCN.....	111
9.4. Wykresy.....	113
9.5. DMX.....	114
9.6. BACnet.....	115
9.6.1. Podgląd zmiennych BACnet.....	115
9.6.2. Urządzenie.....	116
9.6.3. Ustawienia zmiennej.....	116
9.6.4. Stany (tylko wejścia i wyjścia wielostanowe).....	117
9.6.5. Subskrypcje.....	117
9.7. u::Lux.....	118
9.7.1. Komunikacja.....	119
9.7.2. Konfiguracja.....	119
9.8. Shelly.....	122
9.8.1. Komunikacja.....	122
9.8.2. Konfiguracja.....	122
9.8.3. Konfiguracja modułu DOMIQ/Base.....	123
9.8.4. Sterowanie i odczyt stanu.....	123
9.9. IFTTT.....	124
9.10. Grafana i Prometheus.io.....	124
10.0.1. Konfiguracja połączeń.....	125
1. Połączenie.....	125
2. Grupa.....	126
3. Master i Slave.....	127
12.1. Zmienne.....	132
12.1.1. MEM.....	132
12.1.2. VAR.....	133
12.1.3. Zmienne systemowe.....	133
12.2. LCN.....	134
12.2.1. LCN.output.....	136
12.2.2. LCN.outputs.....	138
12.2.3. LCN.relay.....	139
12.2.4. LCN.relays.....	140
12.2.5. LCN.sensor.....	141
12.2.6. LCN.motor.....	142
12.2.7. LCN.motors.....	143
12.2.8. LCN.regulator.....	144

12.2.9. LCN.value .....	146
12.2.10. LCN.variable .....	147
12.2.11. LCN.threshold .....	148
12.2.12. LCN.threshold .....	149
12.2.13. LCN.key .....	150
12.2.14. LCN.sendkey.....	151
12.2.15. LCN.scene.....	151
12.2.16. LCN.scenes.....	152
12.2.17. LCN.transponder .....	154
12.2.18. LCN.dali .....	155
12.2.19. LCN.locks .....	156
12.2.20. LCN.text .....	156
12.2.21. LCN.groups .....	157
12.3. IDS .....	158
12.3.1. IDS.input .....	159
12.3.2. IDS.output .....	159
12.3.3. IDS.armed .....	161
12.3.4. IDS.entry .....	162
12.3.5. IDS.exit.....	162
12.3.6. IDS.alarm.....	163
12.4. DMX .....	164
12.4.1. DMX.<kanal> .....	164
12.4.2. DMX.scene .....	167
12.5. MODBUS .....	168
12.6. EXP .....	169
12.6.1. EXP.output.....	170
12.6.2. EXP.outputs.....	171
12.6.3. EXP.group.....	172
12.6.4. EXP.all .....	173
12.6.5. EXP.input.....	173
12.6.6. EXP.shutter .....	174
12.6.7. EXP.button.....	176
12.6.8. EXP.imp.....	176
12.6.9. EXP.scan.....	177
12.7. DALI .....	178
12.7.1. DALI.1.evg .....	179
12.7.2. DALI.1.group.....	183
12.7.3. DALI.1.all .....	183
12.8. SHELLY .....	184
12.8.1. 1/1PM/1L .....	185
12.8.2. 2.5.....	186
12.8.3. Plug/PlugS.....	188
12.8.4. Dimmer 1/2.....	189
12.8.5. RGBW2 (tylko tryb White) .....	190
12.8.6. Uni.....	191
12.8.7. TRV .....	192
12.8.8. Vintage.....	193
12.8.9. H&T .....	193
12.8.10. Smoke.....	194
12.8.11. Flood.....	194
12.8.12. Door/Window 1/2 .....	194
12.8.13. Motion.....	195
12.8.14. Sense.....	195
12.8.15. Button1 .....	196
12.8.16. i3.....	196
12.8.17. Gas .....	197
12.8.18. EM .....	198
12.8.19. 3EM.....	199

12.8.20. Plus 1/1 PM .....	200
12.8.21. Plus 2 PM .....	201
12.8.22. I4.....	204
12.8.23. Pro 1/ Pro 1PM .....	205
12.8.24. Pro 2 .....	206
12.8.25. Pro 2 PM.....	207
12.8.26. Pro 4 PM.....	207
12.9. ULUX.....	208
12.9.1. ULUX.key.....	209
12.9.2. ULUX.sound .....	210
12.9.3. ULUX.screensaver .....	211
12.9.4. ULUX.page.....	211
12.9.5. ULUX.led .....	212
12.9.6. ULUX.text.....	213
12.9.7. ULUX.volume.....	213
12.9.8. ULUX.brightness .....	214
12.9.9. ULUX.motion .....	214
12.9.10. ULUX.temperature .....	215
12.9.11. ULUX.humidity .....	215
12.9.12. ULUX.CO2.....	216
12.9.13. ULUX.input1 .....	216
12.9.14. ULUX.input2 .....	217
12.10. IFTTT .....	218
12.10.1. Komenda IFTTT.....	218
12.10.2. Zdarzenie IFTTT .....	218
12.11. Grafana.....	219
12.12. UAV .....	220
12.13. Połączenia szeregowo .....	223
12.13.1. LC.SER.line.....	223
12.13.2. LC.SER.config .....	225
12.13.3. LC.SER.send .....	225
12.14. Komunikacja sieciowa .....	226
12.14.1. TCP .....	226
12.14.2. UDP .....	226
12.14.3. HTTP.get.....	227
12.15. Notyfikacje.....	228
12.15.1. REMOTE.message.....	228
12.15.2. REMOTE.notify.....	229
12.15.3. EMAIL.....	230
12.15.4. DISPLAY.screen .....	230
12.15.5. DISPLAY.layer .....	230
12.16. Łączy .....	232
12.16.1. Komendy i zdarzenia sieciowe .....	232
12.16.2. NET .....	233
12.16.3. Połączenia master-slave (MNT) .....	234
12.17. Zdarzenia systemowe.....	235
12.17.1. Restart systemu .....	235
12.18. Identyfikatory własne (makra).....	236





# Rozdział 1

## Wstęp

Niniejsza instrukcja opisuje konfigurację i działanie modułu **DOMIQ/Base**. Czytając tę instrukcję poznasz wszystkie funkcje oferowane przez moduł **DOMIQ/Base** wraz z przykładami zastosowania.

Konfigurację systemu **DOMIQ** można podzielić na trzy główne etapy:

- **Ustawienia parametrów modułu Base**. - Na tym etapie skonfigurujesz najbardziej elementarne ustawienia systemu, takie jak: konfiguracja sieci, dane dostępowe, parowanie z aplikacjami sterującymi, ustawienia integracji z zewnętrznymi urządzeniami podsystemami, itd.
- **Konfiguracja interfejsów użytkownika** - Interfejs użytkownika w aplikacjach sterujących **DOMIQ** jest w pełni konfigurowalny. To bardzo istotny etap, od którego zależy ergonomia pracy z systemem **DOMIQ**.
- **Automatyzacja** - To nieco bardziej złożony proces polegający na dodaniu do instalacji tzw. inteligentnych funkcji. Zaliczają się do nich reguły warunkowe, reguły czasowe oraz skrypty.

**WAŻNE!** Konfiguracja urządzenia i implementacja standardowych funkcjonalności nie wymaga żadnej wiedzy programistycznej. Edytor skryptów w zakładce **Logika** pozwala na realizację bardziej zaawansowanych scenariuszy sterowania i wymaga znajomości języka skryptowego Lua. Dopelnienie tej instrukcji stanowi ulotka instalacyjna modułu **DOMIQ/Base** oraz samouczki prezentujące gotowe do użycia przykłady zastosowań modułu **Base** i innych urządzeń **DOMIQ**.

Samouczki można pobrać z naszej strony internetowej [www.domiq.pl](http://www.domiq.pl), dział **Samouczki**.

### LEGENDA

W instrukcji napotkasz kilka charakterystycznych stylów formatowania tekstu i oznaczeń:

- **DOMIQ/Base** - W ten sposób oznaczane są wszystkie produkty **DOMIQ**.
- **Tylko sieć lokalna** - Ten styl formatowania jest przeznaczony dla wszystkich elementów interfejsu np. nazwy zakładek, nazwy kontrolek, nazwy pól do uzupełnienia itd. oraz nazw własnych zawartych na naszej stronie internetowej.
- *admin, 10, 200* - Tym stylem formatowania oznaczone są wszelkie treści, które ręcznie musi wpisać użytkownik
- `C.LCN.output.0.10.1=on` - W ten sposób reprezentowane są wszystkie wyrażenia związane z identyfikatorami (stan, zdarzenia, komendy).



Tak oznaczane są przykłady.

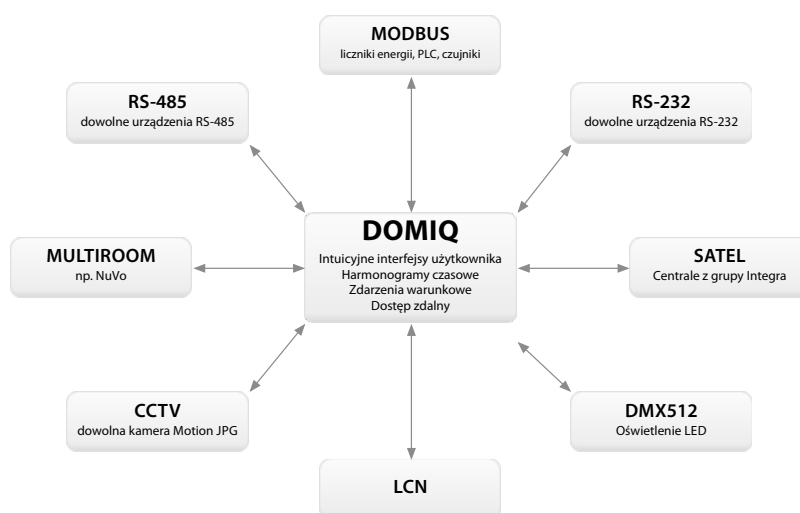
- Przykład 1
- Przykład 2



# Rozdział 2

## Produkty DOMIQ

DOMIQ jest systemem integracyjnym automatyki budynkowej, który pozwala na połączenie wielu często występujących podsystemów inteligentnego budynku w jeden spójny system. Dzięki integracji możliwa jest wymiana informacji między podsystemami, co pozwala na konfigurację lepszych algorytmów sterujących oraz wykorzystanie wspólnych interfejsów użytkownika na urządzeniach mobilnych, panelach dotykowych i komputerach. Dzięki możliwości definiowania nieograniczonej liczby zdarzeń warunkowych i reguł czasowych system DOMIQ znacznie ułatwia proces automatyzacji budynku. Dostęp zdalny zwiększa elastyczność pracy instalatora oraz umożliwia użytkownikowi podgląd stanu budynku z dowolnego miejsca na świecie.



Rys 2.1: Schemat integracji podsystemów automatyki domowej z udziałem DOMIQ

Najważniejsze produkty to:

- **DOMIQ/Base**: moduł sterujący, podwalający na podłączenie do instalacji LCN i sieci Internet
- **DOMIQ/Remote**: aplikacja dla urządzeń mobilnych do sterowania systemem

Do integracji z innymi systemami służą dodatkowe moduły:

- **DOMIQ/Serial-2SI**: integracja z centralami alarmowymi SATEL Integra®
- **DOMIQ/Serial-4DX**: sterowanie oświetleniem z wykorzystaniem DMX-512
- **DOMIQ/Serial-4MB**: integracja z urządzeniami wykorzystującymi protokół MODBUS
- **DOMIQ/Serial-2SG**: integracja z dowolnymi urządzeniami łączem RS-232
- **DOMIQ/Serial-4SG**: integracja z dowolnymi urządzeniami łączem RS-485
- **DOMIQ/Light**: sterowanie oświetleniem z wykorzystaniem standardu DALI
- **DOMIQ/Expander**: moduł do integracji ekspanderów Satel INT-IORS oraz INT-ORS.

# 2.1. DOMIQ/Base

DOMIQ/Base jest podstawowym elementem systemu DOMIQ. Jego funkcjonalność obejmuje:

- Wbudowany interfejs do systemu automatyki budynkowej LCN.
- Obsługę interfejsów użytkownika: wbudowany edytor interaktywnych wizualizacji, edytor menu aplikacji na urządzenia mobilne.
- Możliwość wykorzystania modułu Base jako sprzęgu do segmentów instalacji LCN.
- Tworzenie reguł czasowych, które znacznie ułatwią proces automatyzacji budynku.
- Definiowanie zdarzeń warunkowych, które pozwalają realizować bardziej złożone reguły logiczne.
- Wygodny interfejs konfiguracyjny
- Połączenia typu Base-Base, co umożliwi tworzenie rozległych struktur automatyki budynkowej w oparciu o urządzenia DOMIQ.
- Zdalny dostęp dla instalatorów LCN przy użyciu LCN-PRO (protokół LCN-PCHK),
- Wbudowany zegar czasu rzeczywistego, podtrzymywany bateryjnie, z automatyczną synchronizacją czasu z atomowymi serwerami czasu przez Internet, automatyczna zmiana czasu letni/zimowy.
- Wsparcie protokołów internetowych: IPv4, IPv6, DHCP, AutoIP, DNS, HTTP, SNTP, FTP, Telnet, MQTT.
- Integracja włączników u::Lux.
- Integracja z serwisem IFTTT
- Edytor skryptów Lua, który oprócz standardowych modułów języka zawiera m.in. moduł asynchronicznych socketów TCP i UDP oraz asynchronicznych timerów.
- Możliwość tworzenia własnych maszyn wirtualnych Lua.

## 2.2. DOMIQ/Serial

Moduły **DOMIQ/Serial** są produkowane w dwu wersjach różniących się interfejsem elektrycznym:

- **RS-232** - do komunikacji na krótkie odległości między parą urządzeń
- **RS-485** - do komunikacji na dłuższe odległości oraz komunikacji wielu urządzeń

Poszczególne rodzaje modułów różnią się też oprogramowaniem, które zawsze dostosowywane jest do konkretnego protokołu komunikacyjnego.

Moduły w wykonaniu **RS-485** od wersji R9 posiadają zworkę, która pozwala na odłączenie wbudowanego terminatora, przez co możliwe jest jednoczesne połączenie wielu modułów (np. **D-S4SG-1**) lub też instalacja modułu "w środku" magistrali **RS-485**.

Typ	Opis	Rodzaj	Identyfikatory
<b>D-S2SI-1</b>	Interfejs do integracji z <b>SATEL Integra®</b>	RS-232	IDS
<b>D-S4MB-1</b>	Interfejs master <b>MODBUS</b>	RS-485	MODBUS
<b>D-S4DX-1</b>	Interfejs master <b>DMX-512</b>	RS-485	DMX
<b>D-S2SG-1</b>	Uniwersalny interfejs <b>RS-232</b>	RS-232	SER
<b>D-S4SG-1</b>	Uniwersalny interfejs <b>RS-485</b>	RS-485	SER

Obsługa poszczególnych modułów jest wykonywana poprzez identyfikatory stanu, komend i zdarzeń, które są dedykowane dla każdego typu modułu.

## 2.3. DOMIQ/Light

Moduł **DOMIQ/Light** umożliwia sterowanie instalacjami oświetleniowymi opartymi o standard DALI.

Zgodnie ze standardem DALI, do modułu **DOMIQ/Light** można podłączyć maksymalnie 64 urządzenia DALI. Sterownik zapewnia pełną dwukierunkową komunikację. Podstawowe funkcje modułu Light:

- algorytm automatycznego adresowania opraw (lampy są adresowane jednocześnie i automatycznie)
- informacja o aktualnej jasności i ewentualnej awarii balastu.
- sterowanie pojedynczym lub jednocześnie wszystkimi balastami DALI.
- przypisywanie balastów do grup (każdy balast może przynależeć maksymalnie do 16 grup).
- zapisywanie i ładowanie scen świetlnych (każdy balast może mieć maksymalnie 16 scen).
- zmiana parametrów pracy balastu takich jak: jasność minimalna, maksymalna, rampa, skok jasności.

## 2.4. DOMIQ/Expander

Moduł **DOMIQ/Expander** został zaprojektowany z myślą o integracji i sterowaniu ekspanderów marki SateL. Umożliwia podłączenie maksymalnie 32 ekspanderów INT-IORS lub INT-ORS. Przy użyciu ekspanderów INT-IORS umożliwia zbudowanie systemu z aż 256 wejściami i wyjściami binarnymi. Do wejść można podłączyć przyciski dzwonekowe lub bistabilne, czujki ruchu, kontaktrony i inne urządzenia, które są w stanieysterować wejścia ekspandera. Do wyjść można podłączyć dowolne urządzenia np. światła, rolety, obwody gniazdek elektrycznych, itd.

# 2.5. DOMIQ/Remote

DOMIQ/Remote to bezpłatna aplikacja mobilna do sterowania system automatyki opartym na urządzeniach DOMIQ. Aplikacja jest dostępna dla systemów iOS oraz Android.

Sterowanie i podgląd stanu automatyki budynkowej może być realizowany na dwa sposoby: poprzez wygodne menu, którego zawartość jest w pełni konfigurowalna lub interaktywne wizualizacje.

# Rozdział 3

## Pierwsze kroki

**WAŻNE:** Jeżeli przystępujesz do reedycji konfiguracji modułu Base wykonaj zawsze kopię zapasową, aby później bezproblemowo móc przywrócić poprzednie ustawienia.

W tym rozdziale poznasz niezbędne czynności, aby po raz pierwszy uruchomić system DOMIQ.

### 3.1. Podłączenie

Przewód neutralny należy podłączyć do niebieskiego zacisku (N), przewód fazowy do zacisku szarego (L). Jeżeli Twoja instalacja jest wyposażona w magistralę LCN, wówczas przewód magistrali LCN należy podłączyć do zacisku pomarańczowego (D).

**UWAGA:** Jeżeli magistrala LCN jest podłączona do modułu Base oraz instalacja LCN jest zasilana z różnych faz, należy korzystać z modułów separacji galwanicznej LCN-IS, aby uchronić moduł Base przed uszkodzeniem w przypadku omyłkowego odłączenia przewodu neutralnego.

Złącze **EXT** służy do komunikacji z modułami rozszerzeń systemu.

Moduł powinien również być podłączony do sieci lokalnej za pomocą kabla Ethernet ze złączem RJ-45. Podłączenie modułu do sieci jest niezbędne do jego konfiguracji i sterowania przy użyciu aplikacji mobilnej DOMIQ/Remote.

Wskazane jest aby moduł DOMIQ/Base miał dostęp do sieci Internet. Dostęp ten jest niezbędny do:

- Zdalnego (poza zasięgiem lokalnego WiFi) sterowania systemem.
- Pobierania aktualizacji oprogramowania DOMIQ/Base.
- Synchronizowania zegara czasu rzeczywistego z internetowymi serwerami czasu za pomocą protokołu SNTP.

Po uruchomieniu moduł Base pobiera adres IP z dostępnego w sieci lokalnej serwera DHCP. Jeśli nie jest to możliwe, wówczas przydzielany jest automatycznie adres w standardzie AUTOIP.

## 3.2. Pierwsze logowanie

DOMIQ/Base posiada wbudowany interfejs konfiguracyjny, dalej zwany **konfiguratorem**. Do jego działania niezbędna jest przeglądarka internetowa obsługująca HTML5 oraz język JavaScript w edycji co najmniej ES6.

Wszystkie zmiany w konfiguracji dokonywane w poszczególnych zakładkach konfiguratora są zapamiętywane lokalnie w przeglądarce do momentu naciśnięcia przycisku **Zapisz** w danej zakładce.

Niektóre zakładki konfiguratora posiadają przycisk **Przywróć**. Naciśnięcie tego przycisku powoduje przywrócenie ostatnio zapisanej konfiguracji.

### 1. Wyszukiwanie adresu IP modułu DOMIQ/Base

Do uruchomienia konfiguratora niezbędne jest poznanie adresu IP modułu Base. Można to zrealizować na dwa sposoby:

#### Z użyciem aplikacji DOMIQ/Discover (tylko system Windows)

Aplikacja DOMIQ/Discover, dostępna do pobrania na stronie [www.domiq.pl](http://www.domiq.pl), wykrywa wszystkie moduły DOMIQ/Base podłączone do danej sieci lokalnej.

Ponadto aplikacja Discover umożliwia rejestrację logów z pracy modułu DOMIQ/Base. Aby uruchomić okno logów należy kliknąć dwukrotnie na adresie IP modułu Base. Komunikaty z konsoli mogą być przydatne przy rozwiązywaniu problemów.

#### Przy użyciu statystyk routera

Zaloguj się do swojego routera i sprawdź listę podłączonych urządzeń sieciowych. Numer seryjny modułu Base (dostępny na frontowej etykiecie obudowy) stanowi jednocześnie ostatnich 6 cyfr adresu MAC karty sieciowej modułu Base.

#### Z użyciem wiersza poleceń (dowolny system)

W oknie wiersza poleceń wpisz komendę: `arp -a`. W wynikach należy wówczas odszukać adres MAC modułu Base, który zaczyna się od `00 : 40 : 9D`.

### 2. Uruchomienie konfiguratora

Po wprowadzeniu adresu IP modułu w przeglądarce internetowej, załadowany zostanie ekran logowania. Domyślne dane logowania: `admin/admin` dla dostępu administracyjnego oraz `user/user` dla dostępu dla użytkownika (tylko tryb wizualizacji).

Po naciśnięciu przycisku **Konfiguracja** nastąpi zalogowanie do interfejsu konfiguracyjnego. Konfigurator ma budowę zakładkową. Poszczególne zakładki zostaną opisane w dalszej części instrukcji.

Po naciśnięciu przycisku **Wizualizacja** zostanie wyświetlony ekran wizualizacji umożliwiający sterowanie systemem DOMIQ z poziomu wizualizacji. Opcja **Wizualizacja pełnoekranowa** pozwala na wyświetlenie wizualizacji wypełniając całe okno przeglądarki. W tym trybie rozmiar wizualizacji może być płynnie sterowany przez zmianę rozmiaru okna przeglądarki.



## 3.3. Ekran startowy

Po zalogowaniu w interfejsie konfiguracyjnym wyświetlony zostanie ekran startowy. W głównej mierze zawiera on informacje o kondycji modułu oraz przydatne linki i dane kontaktowe.

Ponadto zawiera dwie opcje dotyczące stanu oprogramowania:

- **Automatycznie sprawdzaj stan aktualizacji** - Jeżeli ta opcja jest włączona, po zalogowaniu moduł sprawdza stan dostępnych aktualizacji systemu. Wyłączenie tej opcji przyspiesza dostęp do interfejsu po zalogowaniu.
- Stan oprogramowania może mieć jedną z trzech wartości:
  - Aktualne* - Twój system jest aktualny.
  - Wymaga aktualizacji* - Dostępne są nowsze wersje oprogramowania. Status w tym przypadku staje jednocześnie linkiem do okna **Aktualizacje**.
  - Błąd pobierania stanu aktualizacji* - Wyświetlany jest, gdy moduł **Base** nie był w stanie pobrać stanu aktualizacji.

## 3.4. Ustawienie danych dostępowych

Po pierwszym zalogowaniu do konfiguratora zalecana jest zmiana loginów i haseł dostępu. W celu zmiany haseł i loginów należy wybrać zakładkę **Użytkownicy**. Zakładka podzielona jest na trzy sekcje: **Dostęp dla użytkownika**, **Dostęp administracyjny** oraz **Zdalny LCN**.

Rys 3.1: Okno ustawień dostępu do modułu DOMIQ/Base

### 1. Dostęp dla użytkownika

W tej sekcji konfigurowane są opcje dostępu dla użytkownika. Użytkownik ma dostęp jedynie do wizualizacji. Z uprawnieniami użytkownika nie jest możliwe zalogowanie się do części konfiguracyjnej. Następujące parametry mogą być modyfikowane:

- **Dostęp**: Tryb dostępu do wizualizacji, możliwe są następujące wartości:
  - **Tylko sieć lokalna** – dostęp do wizualizacji możliwy tylko dla komputerów przyłączonych do tej samej sieci lokalnej, co DOMIQ/Base.
  - **Cały Internet** – dostęp do wizualizacji jest możliwy zarówno z sieci lokalnej jak i przez Internet.
- **Użytkownik**: Identyfikator użytkownika, który należy wpisać aby uzyskać dostęp do wizualizacji.
- **Hasło**: Hasło użytkownika, które należy wpisać aby uzyskać dostęp do wizualizacji.

### 2. Dostęp administracyjny

W tej sekcji konfigurowane są parametry dostępu do interfejsu konfiguracyjnego. Następujące parametry mogą być modyfikowane:

- **Dostęp**: Tryb dostępu do interfejsu konfiguracyjnego możliwe są następujące wartości:
  - **Tylko sieć lokalna** – dostęp do konfiguratora jest możliwy tylko dla komputerów przyłączonych do tej samej sieci lokalnej, co DOMIQ/Base.
  - **Cały internet** – dostęp do interfejsu konfiguracyjnego jest możliwy zarówno z sieci lokalnej jak i przez Internet.
- **Użytkownik**: Identyfikator użytkownika, który należy wpisać aby uzyskać dostęp do konfiguratora.
- **Hasło**: Hasło użytkownika, które należy wpisać, aby uzyskać dostęp do konfiguratora.

### 3. Pierwsze kroki

**W celu zdalnej konfiguracji systemu zalecamy utworzenie połączenia VPN.** W przypadku, gdy jest to niemożliwe, wówczas należy w konfiguracji routera przekierować port o numerze zgodnym z ustawieniem portu WEB konfiguratora - domyślnie 80 (zakładka **Ustawienia**).

### 3. Programowanie LCN

Moduł DOMIQ/Base może pełnić rolę sprzęgu do systemu LCN. W tej sekcji konfigurowane są parametry dostępu do magistrali LCN dla aplikacji LCN-PRO. Następujące parametry mogą być modyfikowane:

- **Dostęp:** Tryb dostępu dla LCN-PRO, możliwe są następujące wartości:
  - **Wyłączone** – dostęp wyłączony.
  - **Tylko sieć lokalna** – dostęp możliwy tylko dla komputerów przyłączonych do tej samej sieci lokalnej co DOMIQ/Base.
  - **Cały internet** – dostęp możliwy zarówno z sieci lokalnej, jak i spoza niej.
- **Użytkownik:** Identyfikator użytkownika, który należy wpisać w LCN-PRO.
- **Hasło:** Hasło użytkownika, które należy wpisać w LCN-PRO.

# Rozdział 4

# Ustawienia

Zakładka **Ustawienia** umożliwia skonfigurowanie podstawowych parametrów pracy modułu. Wszystkie zmiany wykonane w zakładce **Ustawienia** są pamiętane lokalnie w przeglądarce i nie mają wpływu na działanie systemu DOMIQ. Dopiero naciśnięcie przycisku **Zapisz** powoduje zapisanie zmian do modułu **DOMIQ/Base**. Zmiany wprowadzone w zakładce **Ustawienia** są widoczne dopiero po restarcie modułu Base.

Przycisk **Odtwórz** przywraca zawartość zakładki **Ustawienia** do stanu, jaki jest aktualnie zapisany w module.

Zakładka **Ustawienia** podzielona jest na cztery części:

1. Ustawienia ogólne
2. Ustawienia sieciowe
3. Parowanie aplikacji
4. Integracje

# 4.1. Ustawienia ogólne

Dostępne są następujące opcje:

- **System:**
  - **Nazwa** - Identyfikator modułu DOMIQ/Base, który jest wyświetlany na ekranie wyboru modułu Base w aplikacji Remote. Wartość domyślna: *DOMIQ*
  - **Restart** - Przycisk inicjujący ponowne uruchomienie modułu Base.
  - **Ustawienia Fabryczne** - Przycisk inicjujący procedurę przywrócenia modułu Base do ustawień fabrycznych. Przed rozpoczęciem tej procedury pamiętaj o skopiowaniu unikalnych kluczy przypisanych do modułu np. klucz do BACnet, IFTTT i inne, ponieważ podczas przywrócenia modułu do ustawień fabrycznych zostaną one skasowane. Użycie tej opcji spowoduje również skasowanie zmiennych nieulotnych MEM. **Dodatkowe licencje dla aplikacji mobilnych przypisanych do modułu Base nie są kasowane.**
  - **Aktualizacje** - Po naciśnięciu przycisku zostanie wyświetlone okno umożliwiające wykonanie aktualizacji oprogramowania modułu Base oraz podłączonych do niego urządzeń peryferyjnych.
    - Kolumna **Moduł** wyświetla nazwy komponentów oprogramowania. W kolumnie **Aktualna wersja** wyświetlana jest aktualnie zainstalowana wersja oprogramowania.
    - W kolumnie **Dostępna wersja** wyświetlana jest najnowsza wersja oprogramowania dostępna na serwerach DOMIQ. Jeżeli kolumna **Dostępna wersja** jest pusta może to sugerować brak dostępu do internetu (często jest to sygnał błędnych ustawień DNS w przypadku ręcznej konfiguracji ustawień sieciowych).
    - Nieaktualne komponenty oprogramowania są domyślnie zaznaczone. Aby rozpocząć aktualizację kliknij przycisk **Aktualizuj zaznaczone**. Po aktualizacji automatycznej moduł samoczynnie się zrestartuje.
    - Moduł **Base** może zostać zaktualizowany również ręcznie (np. w przypadku instalacji, gdzie moduł nie jest podłączony do Internetu). Wówczas należy z witryny DOMIQ pobrać pliki aktualizacji wraz z przypisanymi do nich kluczami aktualizacji. Klucz należy wkleić w polu pod tabelą, a następnie załadować plik aktualizacji, klikając przycisk **Załaduj**. Po aktualizacji ręcznej wymagany jest również ręczny restart modułu.
  - **Data i czas** - okno umożliwiające ustawienie aktualnej daty i czasu w module Base. Jeżeli moduł Base jest podłączony do internetu, wówczas data i czas ustawiane są automatycznie.
- **Kopia zapasowa konfiguracji:**
  - **Zapisz do pliku** - tworzy kopię zapasową konfiguracji. Pamiętaj o stworzeniu kopii zapasowej za każdym razem, kiedy wprowadzasz znaczące zmiany w instalacji.
  - **Odtwórz z pliku** - umożliwia odtworzenie konfiguracji z wcześniej utworzonej kopii.
- **Lokalizacja** - w polach formularza należy uzupełnić dane geolokalizacyjne budynku oraz wybrać z listy strefę czasową. Uzupełnienie tych ustawień jest niezbędne do prawidłowego działania zegara astronomicznego.

## 4.2. Ustawienia sieciowe

- **Sterowanie przez Internet** - włączenie tej opcji jest niezbędne, aby możliwe było sterowanie zdalne przy użyciu aplikacji mobilnej. Włączenie tej opcji jest również niezbędne, aby móc otrzymywać notyfikacje push wysyłane przez moduł Base. Jeżeli opcja jest włączona przy połączeniach zdalnych pośredniczy serwer DOMIQ (zalecane).

**Uwaga!** Jeżeli chcesz pominąć serwer DOMIQ przy połączeniach zdalnych, wówczas należy skonfigurować połączenie ręcznie w aplikacji mobilnej (niezalecane). Niezbędne jest wówczas posiadanie stałego adresu IP, pod którym dostępny jest moduł Base, a także konieczne jest przekierowanie portu: TCP 4556 dla aplikacji DOMIQ/Remote na iOS  
UDP 44544 dla DOMIQ/Remote na Android.

- **Ukryty** - włączenie tej opcji sprawia, że moduł Base nie jest wykrywany przez aplikację mobilną.
- **DHCP**: Jeśli ta opcja jest zaznaczona, wówczas DOMIQ/Base próbuje automatycznie pobrać ustawienia IP z serwera DHCP (najczęściej router pełni funkcję serwera DHCP). Przy wybraniu tej opcji adres IP zostanie przydzielony automatycznie. Zalecamy korzystanie z opcji **DHCP**. Ponadto dobrą praktyką jest przydzielenie ręcznego adresu IP w ustawieniach routera na podstawie adresu MAC modułu Base. Pozwala to uniknąć problemów z ustawieniami sieciowymi w przypadku wymiany routera i/lub innych elementów architektury sieciowej. Następne trzy pola są dostępne do edycji tylko jeśli opcja **DHCP** jest wyłączona.
- **Adres IP**: Stały adres IP przydzielony modułowi DOMIQ/Base w sieci LAN. W przypadku statycznego adresu IP zalecamy, aby dokonać również rezerwacji tego adresu w ustawieniach routera - jest to sposób na uniknięcie konfliktów w adresacji IP.
- **Maska podsieci**: Maska podsieci IP.
- **Brama**: Adres IP urządzenia stanowiącego punkt wejściowy do danej części sieci (najczęściej adres IP routera/switcha).
- **Główny DNS**: Adres IP pierwszego serwera nazw.
- **Dodatkowy DNS**: Adres IP dodatkowego serwera nazw.

**Po zmianie ustawień sieciowych moduł należy zrestartować.**

**UWAGA!** W przypadku wprowadzenia błędnej konfiguracji sieciowej moduł może stać się nieosiągalny (interfejs konfiguracyjny oraz sterowanie z poziomu aplikacji). W celu tymczasowego ustawienia przydzielenia adresu z użyciem serwera DHCP należy wykonać poniższą procedurę:

1. Odłącz moduł Base od zasilania.
2. Załóż zworkę na złączu **ISP** zgodnie z poniższym rysunkiem:



Rys 4.1: Montaż zworki na złączu ISP

3. Podłącz zasilanie (zworka założona).
4. Moduł Base otrzyma adres z puli DHCP. Odczekaj około 2 minuty na pełne uruchomienie modułu Base, a następnie zaloguj się do interfejsu konfiguracyjnego i wprowadź prawidłowe ustawienia sieciowe i zapisz zakładkę **Ustawienia**.
5. Odłącz zasilanie.
6. Zdejmij zworkę.
7. Uruchom moduł ponownie.

## 4. Ustawienia

8. **Główny SNTP**: Adres IP podstawowego internetowego serwera czasu.
9. **Dodatkowy SNTP**: Adres IP dodatkowego internetowego serwera czasu.
10. **Port web**: Numer portu w module DOMIQ/Base, na którym działa serwer web udostępniający interfejs konfiguracyjny. Domyślna wartość: **80**
11. **Port PCK**: Port wykorzystywany do komunikacji aplikacji LCN Pro z modulem DOMIQ/Base. Domyślna wartość: **4114**

## 4.3. Parowanie aplikacji

Parowanie aplikacji przeznaczonych do sterowania instalacją jest jedną z pierwszych czynności, jaką należy wykonać po uruchomieniu modułu Base

System DOMIQ może być sterowany przez aplikacje:

- **DOMIQ/Remote** - aplikacja na urządzenia mobilne z systemami iOS (Apple) lub Android.
- **DOMIQ/AirDisplay** - aplikacja na komputery używająca biblioteki Adobe Air.

Procedura parowania aplikacji z modulem Base przebiega następująco:

1. Urządzenie z zainstalowaną aplikacją musi znajdować się w tej samej sieci lokalnej, w której jest moduł Base. **Ze względów bezpieczeństwa nie ma możliwości parowania urządzenia spoza sieci lokalnej!**
2. Uruchom aplikację.
3. Aplikacja domyślnie wyświetla widok wyszukanych/sparowanych modułów Base. Moduły wyszukiwane są automatycznie.
4. Kliknij na nazwie modułu Base, który chcesz sparować.
5. Wyświetlony zostanie ekran z polem do wprowadzenia jednorazowego 4-cyfrowego kodu parowania. Aby wygenerować kod kliknij przycisk **Generuj kod** w zakładce **Ustawienia**. Kod ważny jest tylko przez 15 minut.
6. Po wprowadzeniu kodu aplikacja automatycznie dokończy proces parowania.

Połączenia (zarówno lokalne, jak i zdalne) pomiędzy aplikacją a modulem Base są szyfrowane.

### Autoryzacje dla aplikacji

Okno służy do zarządzania autoryzacjami dla aplikacji sparowanych z modulem Base. Okno zawiera następujące informacje:

- Numer seryjny modułu Base. Ten numer jest niezbędny przy zamawianiu dodatkowych licencji parujących dla aplikacji. W celu zakupu dodatkowych licencji prosimy o kontakt na e-mail: **info@domiq.pl**. W e-mailu koniecznie należy podać numer seryjny modułu DOMIQ/Base.
- W tabeli wyświetlane są informacje o urządzeniach sparowanych z modulem Base.
  - W kolumnie **Nazwa** wyświetlana jest nazwa sparowanego urządzenia.
  - Kolumna **ID** wyświetla unikalny identyfikator sparowanego urządzenia.
  - W kolumnie **Opis** możesz wprowadzić opis ułatwiający Ci identyfikację urządzenia. Jest to szczególnie pomocne przy tworzeniu profili użytkowników.
  - Kolumna **Dodany** zawiera datę parowania urządzenia.
  - W kolumnie **Akcja** znajdują się przyciski **Usuń**, które umożliwiają rozparowanie pojedynczego urządzenia.
  - Przycisk **Usuń wszystkie urządzenia** usuwa wszystkie parowania i wymusza ponowne połączenie modułu Base z serwerem DOMIQ.
- Liczba dostępnych wolnych licencji dla aplikacji. Moduł DOMIQ/Base domyślnie sprzedawany jest z jedną licencją dostępową.
- Pole do wprowadzenia klucza autoryzacyjnego dla zakupionych dodatkowych licencji dostępowych.

## Profile użytkowników

Profile użytkowników służą do sterowania widocznością struktury menu wyświetlanej w aplikacji Remote. Dzięki temu z łatwością ograniczysz dostęp do określonych elementów dla wybranych użytkowników systemu DOMIQ.

**Uwaga: Zapisanie profili spowoduje usunięcie wszystkich niezapisanych zmian w zakładce Remote!**

Tworzenie profilu użytkowników przebiega następująco:

1. Dodaj nowy profil klikając przycisk **Dodaj**
2. Uzupełnij jego nazwę. Nazwa musi być unikalna pośród wszystkich profili.
3. Wybierz urządzenia, które mają być przypisane do tego profilu. Tabela po prawej wyświetla nazwy sparowanych urządzeń na podstawie kolumny **Nazwa** lub **Opis** (jeżeli został nadany).

Dokładny opis użycia profili użytkowników został opisany w samouczku: <https://www.domiq.pl/wiki/profile-uzytkownikow-w-remote/>

# 4.4. Integracje

## LCN

Dostępne są następujące opcje:

- **LCN** - Opcja umożliwiająca wyłączenie interfejsu LCN w module Base. Aby zmiany zostały wprowadzone niezbędny jest restart modułu. Gdy pole jest zaznaczone (domyślnie), wówczas interfejs LCN jest uruchomiony.
- **ID** - Identyfikator (Node ID) w domenie adresowej LCN. Domyślna wartość: **254**.
- **Segment**: Wymagane tylko przy użyciu Base jako sprzęgu segmentowego LCN. Należy tu wpisać numer segmentu, do którego Base stanowi sprzęg.
- **Widoczne moduły**: To pole pozwala na ograniczenie puli modułów LCN, z którymi DOMIQ/Base się komunikuje. Należy je wypełnić tylko wtedy, gdy jest faktyczna potrzeba. W standardowych instalacjach nie jest to z reguły wymagane. Niewłaściwe uzupełnienie tego pola będzie skutkowało częściowym lub całkowitym brakiem komunikacji modułu Base z instalacją LCN.

Dopuszczalne są dwie formy zapisu:

- Numery poszczególnych modułów rozdzielone znakiem przecinka.
- Grupy modułów, np. **1-10, 20-31**.
- **Synchronizacja czasu**: Określa z jaką częstotliwością DOMIQ/Base będzie wysyłał do modułów LCN polecenie synchronizacji czasu. Dostępne opcje:
  - **Off**
  - **5 min**
  - **15 min**
  - **1 godzina**
- **Tryb 200 poziomów** - Ta opcja powinna być zaznaczona w przypadku, gdy moduły LCN zostały zaprogramowane w trybie 200 poziomego ściemniania. W sytuacji, gdy w instalacji występują moduły starszej generacji (nie posiadające 200 poziomowego ściemniania), wówczas należy przeprogramować całą instalację na ściemnianie 50 poziomowe.  
**UWAGA!** W przypadku, gdy moduł LCN jest zaprogramowany na ściemnianie 200 poziomowe, ale opcja **Tryb 200 poziomów** nie jest włączona, komenda **włącz** spowoduje wysterowanie ściemniacza na 25%. To typowy objaw wskazujący na konieczność włączenia tej opcji.
- **Monitoruj pozycję silników** - Włącz jeżeli używasz silników z pozycjonowaniem i chcesz, aby moduł Base regularnie odpytywał moduły LCN o pozycję silników (zwiększone obciążenie magistrali LCN).
- **Zdarzenia z innych segmentów** - To pole należy zaznaczyć, jeżeli Base ma odbierać zdarzenia z innych segmentów LCN (tylko, gdy moduł Base pełni rolę sprzęgu segmentowego). Zaznaczenie tego pola oznacza również, że w zakładce **Stan** będzie widoczny stan modułów z innych segmentów LCN.
- **Pobieraj komentarze z modułów LCN** - Wyłączenie tej opcji znacząco przyspiesza skanowanie magistrali LCN. W przeprowadzonych testach, gdzie testowaliśmy instalację, w której pracowało 170 modułów LCN, po wyłączeniu komentarzy czas skanowania instalacji skrócił się z 30 do 9 minut!
- **Skanuj magistralę LCN** - Przycisk inicjujący ręczne skanowanie magistrali LCN. Ponowne zeskanowanie magistrali jest istotne w przypadku wprowadzenia zmian w konfiguracji modułów LCN.

## Satel

- **Pobieraj komentarze z centrali Satel** - wyłączenie tej opcji przyspiesza pobieranie danych z centrali alarmowej Satel.
- **Satel ETHM** - Włącz tą opcję, aby aktywować integrację z centralą alarmową Satel Integra poprzez moduł ETHM-1 Plus lub ETHM-1.
  - **Klucz licencyjny** - W tym polu wklej zakupiony numer licencyjny.
  - **IP** - Adres IP modułu Satel ETHM-1 Plus/ETHM-1
  - **Port** - Port komunikacyjny dla protokołu integracyjnego Satel. Wartość domyślna: **7094**.



- **Pin** - Pin do centrali. Jego podanie jest niezbędne dla odczytu niektórych danych z centrali m.in informacji o nazwach użytkowników. Ta funkcjonalność jest wykorzystywana do generowania logów o zdarzeniach z centrali.
- **Okres logu zdarzeń** - Określa liczbę dni, dla których pobierane są logi z centrali alarmowej. Wartość domyślna: 2.

### BACnet

- **BACnet** - Włącz, aby uruchomić interfejs BACnet w module Base. Obsługa protokołu BACnet w Base jest oferowana jako dodatkowa licencja.
- **Klucz licencyjny** - W tym polu wklej zakupiony numer licencyjny.

### IFTTT

- **IFTTT** - Włącz, aby uruchomić integrację z serwisem IFTTT.
- **Klucz** - W tym polu wklej Twój unikalny klucz otrzymany z serwisu IFTTT.

Więcej o integracji z serwisem IFTTT przeczytasz w samouczku: <https://www.domiq.pl/wiki/integracja-z-ifttt/>.

### UPnP

Włącz, aby uruchomić interfejs UPnP AV w module Base. Interfejs UPnP w Base umożliwia sterowanie urządzeniami kompatybilnymi ze standardem UPnP, m.in głośnikami SONOS.

Więcej o integracji SONOS przeczytasz w samouczku: <https://www.domiq.pl/wiki/integracja-z-sonos/>.

### u::Lux

Włącz, aby uruchomić interfejs obsługujący włączniki u::Lux. Więcej o włącznikach u::Lux możesz przeczytać w dalszej części tego dokumentu oraz w samouczku: <https://domiq.pl/products/ulux>.

### MQTT

Moduł posiada bazową implementację brokera MQTT zgodnego ze standardem w wersji 3.1.1.

- **MQTT** - Włącz, aby uruchomić funkcję brokera MQTT w Base.
- **Shelly** - Włącz, aby aktywować interfejs do integracji z urządzeniami Shelly. Więcej o integracji z urządzeniami Shelly przeczytasz w dalszej części tego dokumentu oraz w samouczku: <https://www.domiq.pl/wiki/integracja-z-shelly/>.

### Połączenia szeregowo

Ustawienia służą do konfiguracji parametrów transmisji połączeń szeregowych z użyciem modułów Serial-2SG i Serial-4SG. Aktualnie możliwe jest podłączenie równocześnie dwóch modułów tego rodzaju.

Dostępne opcje:

- **Prędkość i format ramki** - Określa prędkość transmisji i format pakietu danych.
- **Limit czasu** - Czas przez jaki moduł oczekuje na otrzymanie odpowiedzi po wysłaniu danych.
- **Rodzaj transmisji:**
  - **Linia** - Dane wysyłane są kończone znakami CR LF oraz moduł oczekuje takiego samego zakończenia w danych odbieranych.
  - **Ramka** - Wybierając ten rodzaj transmisji możliwe jest wskazanie niestandardowych znaków końca ramki danych.



# Rozdział 5

## Stan, komendy i zdarzenia

Zrozumienie zasady działania modułu Base bardzo uprości przyswojenie przykładów i opisów przedstawianych w dalszej części dokumentacji. Zrozumienie terminu *identyfikator* (opis poniżej) jest fundamentalne dla uchwycenia idei konfiguracji i działania całej instalacji DOMIQ.

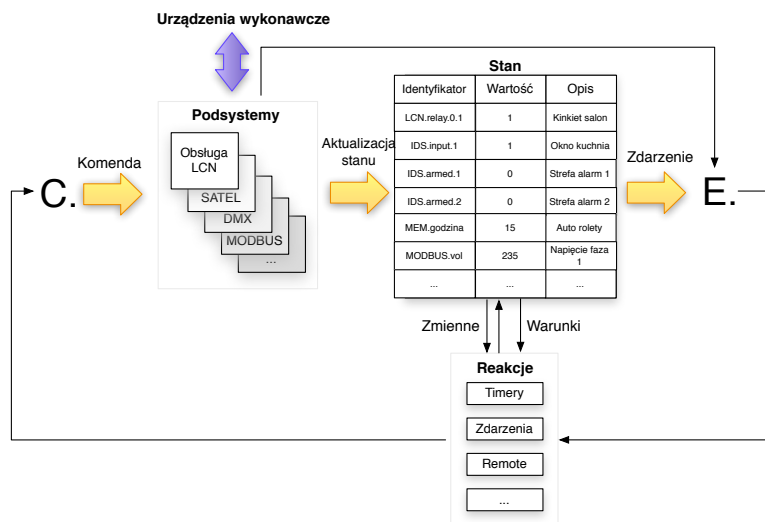
Kilka najważniejszych pojęć to:

- **Podsystem** – to jeden z kilku obsługiwanych interfejsów, np. LCN lub SATEL.
- **Identyfikator** – to ciąg znaków, jednoznacznie identyfikujący urządzenie lub zmienną. Moduł Base posiada całą gamę predefiniowanych identyfikatorów oraz umożliwia definiowanie własnych identyfikatorów przez użytkownika. Identyfikatory wraz z pełnym ich wykazem zostały szczegółowo opisane w rozdziale 12. Identyfikatory. Używając identyfikatora możesz wysłać komendę, zareagować na zdarzenia dotyczące danego identyfikatora oraz wyświetlić jego stan.
- **Stan** – to aktualna wartość identyfikatora (urządzenia/zmiennej).
- **Komenda** – polecenie, które zostanie przesłane do podsystemu, mające za zadanie zmianę stanu urządzenia/zmiennej. Komendy są poprzedzane prefiksem **C** . dodanym do identyfikatora stanu.
- **Zdarzenie** – informacja emitowana przez system DOMIQ w przypadku zaistnienia określonych okoliczności, np. o naruszenie czujnika ruchu, zmiana wartości zmiennej systemowej, wystąpienie alarmu, włączenie przekaźnika, przekroczenie temperatury, itd. Zdarzenia są poprzedzane prefiksem **E** . dodanym do nazwy identyfikatora.

**UWAGA!** Wszystkie pola w interfejsie konfiguracyjnym wymagające prefiksu **C** . lub **E** . zostały oznaczone pomarańczowym tłem i stosowną podpowiedzią. Zastosowanie prefiksów wraz z przykładami zostało opisane w rozdziale 12. Identyfikatory

Przedstawiony diagram obrazuje przepływ informacji w module DOMIQ/Base oraz wzajemne powiązanie **Stanu**, **Komend** i **Zdarzeń** w odniesieniu do przykładowych identyfikatorów.

## 5. Stan, komendy i zdarzenia



Rys 5.1: Przepływ informacji w module DOMIQ/Base

Przestuduj poniższy przykład opisujący przepływ informacji pomiędzy suwakiem w aplikacji **Remote**, a ściemniaczem w systemie **LCN**:



1. Ściemniacze w systemie LCN używają predefiniowanego identyfikatora `LCN.output`. W naszym przykładzie założymy, że jest to identyfikator `LCN.output.0.10.1`, co odpowiada pierwszemu ściemniaczowi w module z ID 10 podłączonym w lokalnym segmencie.
2. Użytkownik w aplikacji DOMIQ/Remote ustawia suwak na połowę zakresu. Do suwaka przypisany jest w/w identyfikator.
3. Wysłana jest komenda do podsystemu LCN, abyysterował żądany obwód świetlny.
4. Urządzenie wykonawcze (w tym przypadku moduł LCN) potwierdza załączenie światła - wysyłany jest pakiet statusowy do modułu Base.
5. Base po przetworzeniu pakietu aktualizuje stan identyfikatora `LCN.output.0.10.1`.
6. Informacja o aktualizacji stanu jest następnie przekazywana do aplikacji Remote, aby wyświetlić bieżącą informację o jasności i prawidłowe położenie suwaka.

I dla uzupełnienia ten sam przykład, ale z użyciem identyfikatorów:



1. Użytkownik w aplikacji DOMIQ/Remote ustawia suwak na połowę zakresu - generuje to komendę `C.LCN.output.0.10.1=50`, która zostaje wysłana do podsystemu LCN.
2. Urządzenie wykonawcze potwierdza załączenie światła - wysyłany jest pakiet statusowy do modułu Base.
3. Po przetworzeniu pakietu statusowego zaktualizowany zostaje stan identyfikatora `LCN.output.0.10.1` oraz wyemitowane jest zdarzenie `E.LCN.output.0.10.1=50`.
4. Informacja o aktualizacji stanu jest następnie przekazywana do aplikacji Remote, aby wyświetlić bieżącą informację o jasności i prawidłowe położenie suwaka.

## Stan

Stany wszystkich identyfikatorów są dostępne w zakładce **Stan** interfejsu konfiguracyjnego. W tej zakładce możesz sprawdzić aktualną wartość każdego urządzenia lub zmiennej dostępnej w systemie DOMIQ. np. aktualną wartość przekaźnika, wyjścia ściemnianego, czujek alarmowych itd.

**WAŻNE!** Moduł Base przechowuje w stanie jedynie wartości niezerowe. Zatem jeżeli dane urządzenie od czasu uruchomienia modułu Base nie zmieniło swojej wartości i ma aktualnie wartość 0, wówczas taki identyfikator nie będzie wyświetlany w tabeli stanu. Jednakże jeżeli ten identyfikator zmieni swą wartość na wartość niezerową, a następnie otrzyma wartość 0, wówczas będzie widoczny w tabeli stanu.

Zakładka **Stan** posiada następujące elementy:

- **Odśwież** - Kliknij ten przycisk, aby pobrać aktualny stan identyfikatorów. Tabela w zakładce **Stan** nie jest automatycznie odświeżana.
- **Usuń** - Służy do usuwania zaznaczonych elementów tabeli.
- Pole wyszukiwania pomaga szybko wyszukać interesujący Cię identyfikator.
- Tabela stanu z aktualnymi wartościami dla poszczególnych identyfikatorów:
  - **Nazwa** - Nazwa identyfikatora
  - **Wartość** - Bieżąca wartość identyfikatora.
  - **Czas od ostatniej zmiany** - Ilość sekund, która upłynęła od ostatniej aktualizacji stanu.
  - **Ostatnia zmiana** - Data i godzina ostatniej zmiany wartości.
  - **Opis** - Opis identyfikatora odczytany z urządzenia źródłowego.

## Zmienne MEM

Moduł Base posiada 8kB pamięci nieulotnej FRAM przeznaczonej na przechowywanie zmiennych. Wartości tych zmiennych są zachowywane podczas restartu urządzenia i przerw w zasilaniu. Dla zwiększenia niezawodności moduł Base zapisuje do pliku co 12h kopię zapasową wartości tych zmiennych.

Zakładka **Stan** posiada zestaw przycisków przeznaczonych do zarządzania zmiennymi MEM.

- **Importuj** - Umożliwia wskazanie pliku kopii zapasowej zmiennych MEM.
- **Eksportuj** - Tworzy kopię zapasową zmiennych MEM w formacie XML i umożliwia jej zapis na dysku komputera.
- **Zapisz do pliku** - Tworzy kopię zapasową, która jest zapisywana w pamięci FLASH modułu Base.
- **Odtwórz z pliku** - Wczytuje wartości zmiennych MEM z kopii zapasowej zapisanej w pamięci FLASH modułu Base.
- **Usuń MEM** - Wykonuje polecenie wyzerowania pamięci FRAM, co powoduje skasowanie wszystkich zmiennych typu MEM.
- **Wyczyść plik kopii zapasowej** - Usuwa plik kopii zapasowej zapisanej w module Base.



# Rozdział 6

## Zasoby

Zakładka **Zasoby** umożliwia wgrywanie nowych plików do modułu Base oraz zarządzanie tymi wcześniej dodanymi. Poznanie tej zakładki przed opisem konfiguracji aplikacji sterujących jest o tyle ważne, że w trakcie konfiguracji interfejsów użytkownika zasoby są powszechnie wykorzystywane.

Okno podzielone jest na trzy części:

1. **Statystyki pamięci**
2. **Struktura plików**
3. **Okno podglądu**

### Statystyki pamięci

W lewym górnym rogu okna wyświetlane jest pole z informacją o aktualnie wolnym i zajęтым miejscu w pamięci modułu DOMIQ/Base.

W lewym dolnym rogu wyświetlane są natomiast informacje o rozmiarze aktualnie wybranego pliku/folderu.

### Struktura plików

W tej sekcji w postaci drzewa wyświetlana jest struktura plików. Pliki zgrupowane są w czterech folderach:

1. Obrazki
2. Ikony
3. Motywy
4. Skrypty

Aby dodać plik(i) należy wybrać folder docelowy. Następnie przecignij plik na **Okno podglądu** (prawa część okna) lub kliknij **Okno poglądu**, aby wybrać plik(i) z dysku.

Aby skasować plik(i), zaznacz go, a następnie kliknij przycisk **Usuń**.

Przycisk **Odznacz wszystko** usuwa zaznaczenie ze wszystkich wybranych plików, niezależnie od przynależności do folderu.

### Obrazki

W tym folderze przechowywane są obrazki, które mogą zostać użyte w wizualizacjach jako tło lub jej części.

Maksymalny rozmiar pliku obsługiwany przez DOMIQ/Base to 250kB.

### Ikony

W tym folderze przechowywane są ikony używane w trybie menu aplikacji Remote. Ikony są używane w przez kontrolki **Strona** oraz **Przycisk**.

## 6. Zasoby

### Motywy

Motywy są używane do graficznego reprezentacji kontrolki na wizualizacji. Motyw to zestaw obrazków, w którym każdy obrazek odpowiada stanowi identyfikatora, który dana kontrolka reprezentuje. Moduł Base ma wgrany fabryczny zestaw motywów, jednakże możesz stworzyć własne pliki motywów, aby dopasować wygląd wizualizacji do własnych preferencji.

Aby bardziej zrozumieć zastosowanie motywów przestuduj przykłady opisane w następnym podrozdziale.

### Tworzenie własnych motywów

Etapy tworzenia motywu:

#### 1. Stworzenie grafiki

Pliki graficzne należy wykonać w dowolnym programie graficznym np. Affinity Photo/Designer, Gimp lub innym. Muszą mieć rozszerzenie `.png` i być zapisane w formacie bez przepłotu. Należy ustawić przezroczystość tła w przypadkach, kiedy zostaje wolny obszar wokół kształtu. W innym przypadku ten obszar zostanie zastąpiony kolorem białym. Unikaj dużych plików - cały plik motywu z reguły waży od kilku do kilkunastu kB.

#### 2. Nadanie nazw plikom wg określonego schematu.

Pliki muszą mieć nazewnictwo w określonej formule: `<nazwa>_<zakres1>_<zakres2>.png` np. `lampaled_1_25.png`, co odpowiada graficznej reprezentacji ikony dla zakresu jasności od 1 do 25%.

#### 3. Spakowanie plików do archiwum bez kompresji.

Można użyć dowolny program do tworzenia archiwów np. winrar, winzip, 7zip itp. W ustawieniach tworzonego archiwum należy uwzględnić dwie bardzo ważne opcje:

- wybrać rodzaj archiwum *ZIP*
- wybrać metodę kompresji jako: *bez kompresji*

W przypadku systemu MacOS można skorzystać z Terminala i wpisać w nim komendę wg poniższego schematu:

```
zip -0 plik.theme plik_0_0.png plik_1_1.png
```

#### 4. Nadanie archiwum nazwy.

Nazwę archiwum należy nadać wedle następującego schematu: `<nazwa>.theme`, gdzie `<nazwa>` musi być tożsama z nazwą użytą do nazywania poszczególnych plików motywu.

#### 5. Wgranie pliku motywu w zakładce **Zasoby**.

Poniżej przedstawiamy dwa przykłady tworzenia motywów. Opis przykładów zakłada, że pliki graficzne są gotowe.

#### Motyw dla ściemniacza

Motyw dla wyjścia ściemnianego obrazowany jest przez zestaw ikon, w którym każda reprezentuje inny stopień wysterowania wyjścia ściemnianego. W tym przykładzie utworzonych zostało pięć ikon, kolejno dla zakresów: 0%, 1-25%, 26-50%, 51-75% oraz 76-100%.



1. Załóżmy, że motyw ma nazwę `lampka`. Wówczas pliki należy nazwać wg nast. schematu: `lampka_0_0.png`, `lampka_1_25.png`, `lampka_26_50.png` itd.
2. Z plików należy utworzyć plik archiwum z rozszerzeniem `.zip`.
3. Archiwum powinno mieć nazwę `lampka.theme`.
4. Gotowy plik należy wgrać do zakładki **Zasoby** -> **Motywy**

#### Motyw dla przełącznika

Motyw dla przełącznika światła definiowany jest analogicznie do przedstawionego wcześniej ściemniacza. Występują dwie różnice:



- W tym przypadku potrzebny jest zestaw tylko dwóch ikon: światło wyłączone i włączone.



- Pliki muszą mieć nadaną nazwę wg następującego schematu: nazwa\_0\_0.png dla światła wyłączonego oraz nazwa\_1\_1.png dla światła włączonego. W przedstawianym przypadku pliki mają następujące nazwy: lampka\_0\_0.png oraz lampka\_1\_1.png.

Pozostałe czynności należy wykonać identycznie, jak w przypadku motywu dla ściemniacza.

### Skrypty

Skrypty to programy napisane w języku programowania Lua, które realizują z góry określoną funkcjonalność. Skrypty mogą być ładowane w zakładce **Logika**.

### Okno podglądu

Prawa część okna **Zasoby**, to **Okno podglądu**. Wyświetlany jest w nim aktualnie zaznaczonego pliku.



# Rozdział 7

## Konfiguracja aplikacji sterujących

Interfejs użytkownika w aplikacjach sterujących DOMIQ jest w pełni konfigurowalny, co pozwala dostosować go do indywidualnych potrzeb klienta. Zatem skonfigurowanie struktury interfejsu jest jedną z najważniejszych czynności w całym procesie konfiguracji systemu DOMIQ.

Oferujemy dwie aplikacje sterujące:

1. **DOMIQ/Remote** - aplikacja na urządzenia mobilne z systemami Apple iOS oraz Android.
  - Aplikacja DOMIQ/Remote może pracować z jednym z dwóch trybów:
    - **Menu** - Ten tryb składa się z list kontrolnych, które umożliwiają sterowanie budynkiem lub wyświetlają aktualny stan urządzeń/zmiennych. Tryb Menu jest domyślnie wyświetlany, gdy urządzenie mobilne znajduje się w orientacji pionowej. Istnieje ponadto możliwość ustawienia trybu Menu dla obu orientacji urządzenia. W tym celu w ustawieniach aplikacji mobilnej należy aktywować opcję **Tylko tryb menu**. Tryb Menu nie jest dostępny w aplikacji AirDisplay. Do konfiguracji tego trybu **Menu** służy zakładka **Remote**.
    - **Wizualizacja** - Tryb wizualizacji to graficzna reprezentacja budynku (tło), na której wyświetlane są aktywne elementy sterujące i wyświetlające stan urządzeń/zmiennych. Tryb wizualizacji umożliwia błyskawiczny podgląd całości budynku/kondygnacji. Dzięki funkcjom nawigacyjnym możliwe jest tworzenie wizualizacji składającej się z wielu ekranów. Do konfiguracji tego trybu służy zakładka **Wizualizacja**.
  - **ZALECENIE:** Jeżeli po raz pierwszy konfigurujesz moduł Base, zalecamy przed przystąpieniem do dalszego etapu, aby zainstalować aplikację DOMIQ/Remote i zapoznać się z wbudowanym trybem demo. Dzięki temu poznasz budowę interfejsu użytkownika i łatwiej będzie Ci zrozumieć omawiane dalej koncepcje. Tryb menu nie wymaga posiadania połączenia z modułem Base. Aplikacja jest dostępna za darmo w AppStore oraz Google Play.
  - **WAŻNE!** Wszystkie sparowane aplikacje używają wspólnej konfiguracji, zatem wyświetlają tę samą strukturę interfejsu. Aby zróżnicować strukturę wyświetlanego Menu, użyj profili użytkowników opisanych wcześniej.
2. **DOMIQ/AirDisplay** - Aplikacja w technologii Adobe Air do zainstalowania na komputerze. Aplikacja wyświetla wyłącznie tryb Wizualizacji.

# 7.1. Remote

Zakładka **Remote** zawiera edytor do tworzenia struktury interfejsu trybu Menu dla aplikacji DOMIQ/Remote. Ekran podzielony jest na dwie części.

Po lewej stronie znajduje się lista dostępnych kontrolkek, które mogą być użyte do tworzenia interfejsu użytkownika. Dostępne elementy zostały podzielone na cztery główne grupy:

1. Struktura
2. Kontrolki kanałowe
3. LCN
4. IDS

W prawej części okna wyświetlana jest struktura interfejsu użytkownika.

- Nowe elementy dodawane są metodą "przeciągnij i upuść". Zatem aby dodać nowy element do struktury, wybierz go z listy dostępnych kontrolkek, a następnie upuść w żądanym miejscu drzewa struktury.
- Aby zmienić pozycję elementu w strukturze, przeciągnij wybrany element i upuść w nowym miejscu.
- Jeżeli element nie może być dodany w wybranym miejscu, wówczas zobaczysz czerwony znacznik. Dla prawidłowych operacji znacznik ma kolor zielony.
- W drzewie struktury obowiązuje kilka zasad dotyczących hierarchii elementów:
  - Na głównym poziomie można dodać tylko elementy typu **Strona**.
  - Bezpośrednio wewnątrz **Stron** można dodawać wyłącznie elementy typu **Sekcja**.
  - Wewnątrz **Sekcji** mogą znajdować się wszystkie dostępne kontrolki, z wyjątkiem innej **Sekcji**.
- Aby zmienić właściwości wybranego elementu, należy kliknąć na nim dwukrotnie - wyświetlone wówczas zostanie okno edycji kontrolki.
- Aby usunąć element(y), należy zaznaczyć go przez kliknięcie w pole wyboru po prawej stronie etykiety, a następnie kliknąć przycisk **Usuń** lub klawisz **backspace** na klawiaturze. Jeśli usuwany element interfejsu ma zagnieżdżone elementy, wówczas one również zostaną usunięte.
- Aby skopiować element, kliknij na nim (po prawej stronie etykiety pojawi się błękitny prostokąt), a następnie kliknij przycisk **Kopiuj** lub kombinację klawiszy **ctrl + c**. Element został skopiowany i znajduje się w schowku.
- Aby wkleić skopiowany element użyj przycisku **Wklej** lub skorzystaj ze skrótu klawiaturowego **ctrl + v**. Skopiowana kontrolka zostanie umieszczona w następującej lokalizacji:
  - poniżej aktualnie wybranej kontrolki;
  - na końcu sekcji, jeżeli zaznaczony jest element typu **Sekcja**;
- Pole wyszukiwania pozwala na szybkie wyszukanie elementów w strukturze. Możliwe jest przeszukiwanie na podstawie etykiety oraz typu kontrolki. Wpisując w pasku wyszukiwania: *ściemniacz*, zostaną wyświetlone wszystkie kontrolki tego typu, występujące w całej konfiguracji.
- Pole wyboru **Zaznacz wszystko** umożliwia szybkie zaznaczenie wszystkich elementów struktury. Jest to szczególnie przydatne, kiedy chcesz szybko usunąć całą strukturę.

W górnej części zakładki znajduje się pasek z przyciskami:

Przyciski **Kopiuj**, **Wklej** oraz **Usuń** zostały już omówione.

- **Importuj** - Umożliwia zaimportowanie pliku .xml z konfiguracją.
- **Eksportuj** - Umożliwia zapis do pliku .xml bieżącej struktury interfejsu.
- **Eksportuj zaznaczone** - Umożliwia zapis do pliku jedynie elementów, które mają zaznaczone pole wyboru. Często sytuacją jest powielanie danej funkcjonalności w kolejnych instalacjach. W ten sposób możesz wyeksportować tylko część interfejsu i użyć jej w innej lokalizacji.
- **Zapisz** - Zapisuje bieżącą konfigurację w pamięci modułu Base.
- **Przywróć** - Wczytuje do drzewa struktury ostatnio zapisaną konfigurację.

W dalszej części tego podrozdziału opisaliśmy wszystkie dostępne kontrolki.

## Struktura

### 1. Strona

**Strona** to podstawowy element struktury trybu Menu w aplikacji DOMIQ/Remote. Głównym widokiem trybu Menu w aplikacji Remote jest siatka ikon - jest to znane i bardzo popularne rozwiązanie stosowane m.in. na pulpitych systemów operacyjnych urządzeń mobilnych. W przypadku aplikacji Remote, elementy siatki są odnośnikami do kontrolki typu **Strona**, które znajdują się na głównym poziomie struktury. W ten sposób użytkownik ma błyskawiczny dostęp do najczęściej używanych stron interfejsu.

O elemencie typu **Strona** należy myśleć, jako o nadrzędnym konterze w którym umieszczane są inne kontrolki. Każda strona odpowiada osobnemu ekranowi w aplikacji. Bezpośrednio do **Strony** można dodać wyłącznie elementy typu **Sekcja**. Następnie kolejne **Strony** mogą być w nich zagnieżdżone, dzięki czemu możliwe jest tworzenie struktur hierarchicznych.

Najczęściej spotykanym podejściem do tworzenia struktury jest podział kontrolki:

- **wg pomieszczeń** - Struktura dzielona jest wg pomieszczeń w budynku. W przypadku takiego podejścia na głównym poziomie struktury umieszczane są strony odpowiadające poszczególnym pomieszczeniom, a wewnątrz nich kontrolki sterujące dostępne w danym pomieszczeniu.
- **wg funkcjonalności** - Na głównym poziomie struktury znajdują się strony grupujące kontrolki sterujące określoną funkcjonalnością budynku, np. oświetlenie, ogrzewanie, rolety, alarm, kamery itd. Wewnątrz danej strony należy następnie umieścić kontrolki sterujące poszczególnymi urządzeniami. Dodatkowo dla lepszej organizacji warto jest zgrupować kontrolki przypisane do danego pomieszczenia wewnątrz elementów typu **Sekcja**.

Oczywiście można stosować podejście łączone.

## Edycja kontrolki

Każda strona musi mieć przypisaną etykietę (pole **Etykieta**). Zalecane jest, aby kontrolki na głównym poziomie miały przypisane ikony, wówczas będą one wyświetlane wewnątrz siatki na ekranie głównym (lista rozwijalna Ikona). Do strony można przypisać listę komend, które zostaną wykonane w przypadku potrząśnięcia urządzeniem mobilnym (tylko system iOS i tylko, gdy strona jest aktualnie wyświetlona) - zakładka **Potrząśnięcie**. Ponadto możliwe jest przypisanie listy komend, które zostaną wykonane, gdy dana strona zostaje wyświetlona w aplikacji - zakładka **Przyciśnięcie**.

Tabela **Akcje** została opisana w podrozdziale 8.4 Akcje.

Tabela **Kontrola dostępu** umożliwia wybór profili użytkowników, którzy mają mieć dostęp do tego elementu. Użytkownicy bez uprawnień nie będą widzieli tego elementu w aplikacji Remote na swoich urządzeniach.

### 2. Sekcja

**Sekcja** jest elementem, który pozwala grupować kontrolki, np. sekcja sterowania oświetleniem, wentylacją itd.

Jak wspomnieliśmy wcześniej, do sekcji można dodawać **Strony**, co umożliwia tworzenie bardziej złożonych struktur interfejsów użytkownika. **Strona** dodana do **Sekcji** jest w aplikacji Remote oznaczona znakiem ostrego nawiasu po prawej stronie etykiety.

## Edycja kontrolki

**Sekcja** może, ale nie musi mieć przypisanej etykiety.

Podobnie jak **Strona** posiada również tabelę **Kontrola dostępu**.

## 7. Konfiguracja aplikacji sterujących

### Kontrolki kanałowe

Kontrolki z tej grupy bazują na identyfikatorach, o których pisaliśmy w rozdziale 5. Stan, komendy i zdarzenia. Szczegółowy opis identyfikatorów znajdziesz w rozdziale 12. Identyfikatory.

#### 1. Grupa przycisków

Ten element interfejsu użytkownika pozwala na grupowanie **Przycisków**. Domyślnym zastosowaniem jest grupowanie przycisków realizujących jakąś wspólną funkcjonalność. Przykładem może być grupa przycisków do sterowania nastawą prędkości wentylacji, gdzie każdy przycisk ma przypisaną komendę z określoną nastawą prędkości. Innym przykładem jest grupa przycisków wywołujących sceny świetlne w pomieszczeniu.

#### Edycja kontrolki

- **Etykieta** - Etykieta będzie wyświetlana w aplikacji nad grupą przycisków.
- **Liczba kolumn** - Określa w ilu kolumnach będą zgrupowane przyciski. Do wyboru są wartości od **1** do **8**. Wybranie **1** oznacza, że przyciski będą wyświetlane jeden pod drugim. Jeżeli wybrana zostanie **2**, wówczas przyciski wyświetlane będą w dwóch kolumnach itd. Domyślna wartość: **4**.

#### 2. Przycisk

**Przycisk** to bardzo uniwersalny element interfejsu użytkownika. Może być częścią **grupy przycisków** lub występować samodzielnie. **Przycisk** różni krótkie i długie naciśnięcie (powyżej sekundy) oraz puszczenie przycisku.

#### Edycja kontrolki

- **Etykieta** - Etykieta wyświetlana będzie w aplikacji Remote obok przycisku.
- **Ikona** - Ikona, która będzie wyświetlana wewnątrz przycisku.
- Do każdego rodzaju akcji można przypisać listę komend - służą do tego zakładki **Krótko**, **Długo** oraz **Puść**. Każda z zakładek zawiera tabelę **Akcję**, która została omówiona w podrozdziale 8.4 Akcje.

#### 3. Kamera

Element **Kamera** służy do wyświetlania obrazu z kamery IP w formacie MJPEG.

#### Edycja kontrolki

- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **URL** - Adres strumienia wideo w formacie MJPEG.
- **Użytkownik** - Nazwa użytkownika do zalogowania się do kamery.
- **Hasło** - Hasło użytkownika do zalogowania się do kamery. Pola **Użytkownik** i **Hasło** są opcjonalne. Należy je uzupełnić jeżeli dostęp do strumienia wymaga autoryzacji.
- **Typ** - Możliwe wartości: *Dowolny MJPEG*, *Vivotek*, *Axis*, *Mobotix*, *Dowolny JPEG*. Jeżeli używasz kamery Axis, Mobotix lub Vivotek w polu **URL** wystarczy podać nazwę hosta i/lub port, nie jest wymagana dalsza część adresu.
- **Obrotowa głowica** - Ta opcja powinna być zaznaczona, jeśli kamera ma obrotową głowicę.
- **Regulacja powiększenia** - Ta opcja powinna być zaznaczona, jeśli kamera posiada i ma mieć sterowaną zmienną ogniskową obiektywu.

Dalsze pola należy wypełnić jedynie dla kamer posiadających sterowaną ruchomą głowicę. Poruszanie palcem po ekranie urządzenia z aplikacją DOMIQ/Remote będzie powodowało ruch głowicy kamery. Treści poleceń należy szukać w instrukcji obsługi kamery.

- **Lewo** - Polecenie dla ruchu kamery w lewo.

- **Prawo** - Polecenie dla ruchu kamery w prawo.
- **Góra** - Polecenie dla ruchu kamery w górę.
- **Dół** - Polecenie dla ruchu kamery w dół.
- **Wąski kąt**: - Funkcja aktualnie nieobsługiwana
- **Szeroki kąt**: - Funkcja aktualnie nieobsługiwana

Więcej o użyciu kamer w aplikacji Remote przeczytasz w samouczku: <https://www.domiq.pl/wiki/kamery-w-remote/>

## 4. Wykres

Kontrolka przeznaczona do wyświetlania wykresów z oprogramowania Grafana.

### Edycja kontrolki

- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Lokalny URL** - Adres URL pod którym wykres dostępny jest w sieci lokalnej.
- **Zdalny URL** - Jeżeli podgląd wykresu ma być dostępny poza siecią lokalną, wówczas w tym polu należy wprowadzić adres tablicy w Grafana, który jest osiągalny z Internetu.

Więcej o integracji z oprogramowaniem Grafana przeczytasz w samouczku: <https://www.domiq.pl/wiki/wykresy-w-domiq-integracja-z-grafana/>

## 5. Przełącznik

**Przełącznik** jest jednym z najczęściej używanych elementów interfejsu, który pozwala realizować sterowanie dowolnego identyfikatora, przełączając jego wartość pomiędzy dwoma stanami. Domyślnie dla stanu włączonego **przełącznik** wysyła wartość *on*, a dla wyłączonego *off*. Wykorzystując przełącznik można sterować np. wyjściem przekaźnikowym, ściemnianym, ustawiać wartości zmiennych (VAR, MEM) itd.

### Edycja kontrolki

- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Kanał** - Nazwa identyfikatora, którym **Przełącznik** ma sterować.



- `LCN.relay.0.36.1`  
Sterowanie przekaźnikiem nr 1 w module LCN o adresie 36.

- **Włącz** (opcjonalnie) - Wartość jaka zostanie wysłana do identyfikatora, gdy przełącznik zostanie włączony.
- **Wyłącz** (opcjonalnie) - Wartość jaka zostanie wysłana do identyfikatora, gdy przełącznik zostanie wyłączony.
- **Parametry** - Dodatkowe parametry, które zostaną wysłane razem z komendą. Jeżeli parametrów ma być przesłanych kilka, należy je rozdzielić średnikiem.



- `ramp:3`  
Jest to przykład użycia parametru do sterowania oświetleniem z rampą. Można zastosować m.in do sterowania wyjściem ściemnianym LCN lub kanałem DMX.
- `pin:12345`  
Jest to przykład dla użycia parametru do sterowania wyjściem w centrali Satel.

- **Odwróć wartość** - Zaznaczając to pole odwrócona zostaje logika działania przełącznika.

## 6. Ściemniacz

**Ściemniacz** przeznaczony jest do sterowania dowolnym identyfikatorem w zadeklarowanym przedziale wartości. Domyślny przedział to 0-100. Ściemniacz może być użyty do sterowania oświetleniem, prędko-

## 7. Konfiguracja aplikacji sterujących

ścią wentylatora, głośnością, temperaturą, itd.

**Ściemniacz** jest wyświetlany w aplikacji Remote jako suwak. Kliknięcie na etykiecie z aktualną wartością realizuje funkcję przełączającą pomiędzy minimalną i maksymalną wartością przypisanego zakresu.

### Edycja kontrolki

- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Kanał** - Nazwa identyfikatora, którym **Ściemniacz** ma sterować.
- **Minimum** - Minimalna wartość na jaką można ustawić suwak. Zakres od -99999 do 99999.
- **Maksimum** - Maksymalna wartość na jaką można ustawić suwak. Zakres od -99999 do 99999.
- **Skok** - Określa minimalną zmianę, jaką można wprowadzić przesuwając suwak. Domyślna wartość: 1. Zakres od -99999 do 99999.



- LCN.output.0.112.1  
Sterowanie wyjściem ściemnianym nr 1 w module LCN o adresie 112.
- UAV.volume.Biuro  
Sterowanie głośnością odtwarzacza Sonos.

## 7. Światło RGBW

Kontrolka **Światło RGBW** jest uniwersalną kontrolką do sterowania oświetleniem RGB(W). Kanał biały (W) jest opcjonalny.

Przykłady zastosowania:

- Sterowanie oświetleniem LED przy pomocy protokołu DMX-512 i modułu Serial-4DX. Więcej o tym rodzaju sterowania przeczytasz w samouczku: <https://www.domiq.pl/wiki/sterowanie-oswietleniem-led/>.
- Sterowanie oświetleniem podłączonym do wyjść ściemnianych modułu LCN.

### Edycja kontrolki

- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Czerwony** - Identyfikator, który odpowiada za sterowanie składową czerwoną.
- **Zielony** - Identyfikator, który odpowiada za sterowanie składową zieloną.
- **Niebieski** - Identyfikator, który odpowiada za sterowanie składową niebieską.
- **Biały** - Identyfikator, który odpowiada za sterowanie składową białą.

Kontrolka **Światło RGBW** wyświetlana w aplikacji Remote składa się z:

1. **Suwak jasności**: Steruje jasnością oświetlenia (wspólna dla wszystkich składowych). Aby wyłączyć oświetlenie należy przesunąć suwak w skrajne lewe położenie. Gdy oświetlenie jest wyłączone **Pierścień wyboru koloru** jest nieaktywny.
2. **Suwak nasycenia**: Steruje nasyceniem koloru oświetlenia.
3. **Pierścień wyboru koloru**: Służy do wyboru kolor z zakresu barw RGB.
4. **Wskaźnik aktualnego koloru**: Wyświetla aktualnie wybrany kolor z uwzględnieniem jasności i nasycenia.
5. **Suwak jasności składowej białej** - Wyświetlany tylko, gdy pole **Biały** jest wypełnione. Steruje jasnością oświetlenia białego.

## 8. Wartość

Kontrolka **Wartość** to uniwersalny element interfejsu służący do ustawiania i wyświetlania wartości. Kontrolka może być użyta do nastawiania i wyświetlania temperatury, zmian wartości zmiennych VAR/MEM itd.

W trybie "Tylko do odczytu" stanowi uniwersalny element do wyświetlania stanu dowolnego identyfikatora w systemie DOMIQ. Dzięki dodatkowym parametrom w ustawieniach kontrolki, możliwe jest ustalenie formatu wyświetlanej wartości.



## Edycja kontrolki

- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Kanał** - Identyfikator, którym chcesz sterować lub wyświetlać stan.
- **Jednostka** - Jednostka wyświetlanej wartości.
- **Tylko do odczytu** - Gdy pole zaznaczone, kontrolka jedynie wyświetla stan identyfikatora, ale bez możliwości zmiany wartości.
- **Minimum** - Minimalna wartość, którą można ustawić. Zakres od -99999 do 99999.
- **Maksimum** - Maksymalna wartość, którą można ustawić. Zakres od -99999 do 99999.
- **Skok** - Określa minimalną zmianę, jaką można wprowadzić. Domyślna wartość: 1. Zakres od -99999 do 99999.
- **Przesunięcie** - Wprowadzona tu wartość będzie dodana do aktualnej wartości identyfikatora. Zakres od -99999 do 99999. Domyślna wartość: 0
- **Mnożnik** - Wartość przez którą pomnożona będzie aktualna wartość identyfikatora. Zakres od -99999 do 99999. Domyślna wartość: 1.
- **Miejsca dziesiętne**: Określa ile miejsc dziesiętnych będzie wyświetlanych. Zakres od 0 do 4. Domyślna wartość: 0.

## 9. Roleta

Uniwersalna kontrolka do sterowania roletami. Kontrolka wysyła komendy: *up*, *down* oraz *stop* do wskazanego identyfikatora.

## Edycja kontrolki

Definicja tego elementu składa się z następujących właściwości:

- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Kanał** - Identyfikator odpowiedzialny za sterowanie roletą. Przykłady identyfikatorów poniżej:



- `LCN.motor.0.36.1`  
Sterowanie roletą z pozycjonowaniem nr 1 podłączoną do modułu LCN o adresie 36, w segmencie 0.
- `EXP.shutter.1.1`  
Sterowanie roletą nr 1 przy użyciu modułu DOMIQ/Expander, podłączoną do modułu SATEL INT-IORS o adresie 1.
- `rolety.parter`  
Sterowanie stanem identyfikatora własnego `rolety.parter`.

## 10. Status

Kontrolka **Status** służy do graficznej reprezentacji stanu dowolnego identyfikatora np. wyjścia lub przełącznika LCN, czujki systemu IDS, zmiennej logicznej itd.

## Edycja kontrolki

- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Kanał** - Identyfikator, którego stan ma być wyświetlony.
- **Ikona wł.:** Graficzna reprezentacja dla stanu włączonego.
- **Ikona wył.:** Graficzna reprezentacja dla stanu wyłączzonego.

## 7. Konfiguracja aplikacji sterujących

### 11. Regulator

**Regulator** jest uniwersalną kontrolką do sterowania temperaturą. Graficzna reprezentacja kontrolki w aplikacji Remote składa się z elementu nastawnego oraz wykresu temperatury mierzonej przez czujnik wskazany w konfiguracji. Wykres przedstawia zmiany temperatury w ostatnich 24h. Dane źródłowe wykresu nie są zapamiętywane po restarcie modułu Base.

#### Edycja kontrolki

- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Kanał z wartością zadaną** - Identyfikator, do którego wysyłane będą komendy z wartością zadaną temperatury.
- **Kanał z wartością aktualną** - Identyfikator czujnika temperatury mierzonej. Na podstawie wartości tego identyfikatora tworzony jest wykres temperatury.
- **Minimum** - Minimalna wartość temperatury zadanej.
- **Maksimum** - Maksymalna wartość temperatury zadanej.

**WAŻNE!** Kontrolka ma wbudowane skalowanie wartości wg skali LCN. Wyraża się ona wzorem:

$$\text{wartość} * 10 + 1000.$$

Zatem aby element nastawny działał prawidłowo oraz aby wykres był kreślony, należy zadbać o to, aby wartości przekazywane do identyfikatorów były wyrażone wg wspomnianej skali.

Poniższe przykłady zawierają pary identyfikatorów, gdzie pierwszy to wartość zadana, a drugi aktualna:



- LCN.regulator.0.10.1  
LCN.value.0.10.t

Wartość zadana będzie przekazywana do/z regulatora LCN nr 1 w module o adresie 10 i w segmencie 0. Wartość bieżąca temperatury będzie odczytywana ze zmiennej t (w nowszych modułach zmienna 1).

**WAŻNE!** W przypadku systemu LCN dla zachowania kompatybilności wstecznej, dopuszczalne jest użycie tylko pierwszych trzech zmiennych i identyfikatora LCN.value. Pierwsze trzy zmienne odpowiadają identyfikatorom: LCN.value.S.M.t, LCN.value.S.M.r1, LCN.value.S.M.r2, gdzie S i M, to segment i moduł.

- MEM.set  
MEM.current

Przykład sterowania z użyciem zmiennych MEM.

### 12. Tekst

Element **Tekst** służy do wyświetlania dowolnej treści tekstowej np. napisy, wartości pomiarowe, stany przekaźników, zmiennych logicznych itd.

#### Edycja kontrolki

- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Kanał** - Identyfikator, którego zawartość chcesz wyświetlić w formie tekstowej.

### 13. Wybór

Kontrolka **Wybór** jest bardzo uniwersalnym elementem, który ma zastosowanie wszędzie tam, gdzie zachodzi potrzeba, aby dać użytkownikowi systemu wybór spośród predefiniowanej listy wartości.

Przykłady zastosowania:

- wybór dni tygodnia, w którym ma być realizowana dana reguła czasowa;
- wybór trybu pracy ogrzewania/klimatyzacji;

- wybór prędkości wentylacji;
- wybór stref, która mają być podlewane;

## Edycja kontrolki

- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Kanał** - Identyfikator z/do którego będzie przesyłana wartość opcji wybranych przez użytkownika.
- **Przynajmniej jeden** - Jeżeli ta opcja jest zaznaczona, użytkownik musi wybrać co najmniej jedną z dostępnych opcji, aby zmienić dotychczasowy stan identyfikatora.
- **Wielokrotny wybór** - Jeżeli aktywne, wówczas użytkownik ma możliwość wyboru wielu opcji z dostępnej listy. Jeżeli użytkownik wybierze kilka opcji, wówczas ich wartości zostaną zapisane w stanie identyfikatora rozdzielone przecinkiem (patrz przykłady poniżej). Jeżeli opcja nie jest zaznaczona, wówczas kontrolka pracuje w trybie jednokrotnego wyboru (jak typowy radio-button).
- **Tylko do odczytu** - Kontrolka wyświetla wybrane wartości na podstawie bieżącego stanu identyfikatora bez możliwości zmiany (ekran wyboru jest niedostępny). Jest to bardzo przydatne wszędzie tam, gdzie zachodzi potrzeba dokonania konwersji z wartości numerycznych na wartości tekstowe - np. z numerów dni tygodnia na ich nazwy lub z numerycznych nastaw prędkości nawiewu na nazwy bardziej przyjazne dla użytkownika.
- **Tabela wartości** - To główny element kontrolki, który służy do definiowania zestawu opcji, które będą wyświetlane na liście wyboru w aplikacji Remote.
  - **Dodaj** - Kliknij aby dodać nowy wiersz w tabeli.
  - **Usuń** - Zaznacz element i kliknij ten przycisk, aby usunąć wiersz(e) w tabeli.
  - **Wartość** - W tej kolumnie wpisz wartość, która zostanie wysłana do identyfikatora, kiedy ta opcja zostanie wybrana przez użytkownika.
  - **Nazwa skrócona** - Wartość tu wprowadzona będzie wyświetlana obok etykiety na liście kontrolki w menu Remote.
  - **Nazwa pełna** - Wartość tu wprowadzona będzie wyświetlana w widoku wyboru w aplikacji Remote.

Poniżej przedstawiamy przykłady konfiguracji kontrolki **Wybór** z rzeczywistych realizacji.

### 1. Wybór dni tygodnia, w którym ma być realizowane zdarzenie czasowe.

Uwaga! Pierwszy dzień tygodnia to niedziela.

**Kanał:** *MEM.days*

**Wielokrotny wybór:** *zaznaczone*

Wartość	Nazwa skrócona	Nazwa pełna
1	Nd	Niedziela
2	Pn	Poniedziałek
3	Wt	Wtorek
4	Śr	Środa
5	Cz	Czwartek
6	Pi	Piątek
7	Sb	Sobota

Założmy, że użytkownik wybrał dni od poniedziałku do piątku. Wówczas identyfikator *MEM.days* będzie miał wartość: *2,3,4,5,6*.

Następnie identyfikator *MEM.days* możemy wykorzystać w definicji timera, jako źródło danych dla dni tygodnia, w którym timer ma być wywoływany.

Przykład definicji takiego timera znajdziesz na końcu rozdziału: 8.3 Timery.

### 2. Wybór biegu wentylacji.

Integracja z systemami wentylacji najczęściej odbywa się z użyciem protokołu MODBUS i wówczas sterowanie prędkością odbywa się poprzez zmianę wartości rejestru MODBUS w centrali wentylacyjnej. Więcej o integracji z użyciem protokołu MODBUS dowiesz się z rozdziału: **9.2 MODBUS**

**Kanał:** *MODBUS.int.vent.speed*

**Wielokrotny wybór:** *odznaczone*

**Przynajmniej jeden:** *zaznaczone*

## 7. Konfiguracja aplikacji sterujących

Wartość	Nazwa skrócona	Nazwa pełna
1	Off	Wyłączony
2	Low	Wolno
3	Mid	Średnio
4	High	Szybko
5	Full	Pełna prędkość

Wartość przypisana do wybranej przez użytkownika opcji zostanie wysłana jako komenda do identyfikatora wpisanego w polu **Kanał**, w ten sposób zmieniając prędkość wentylacji.

## 14. Czas

Kontrolka **Czas** to element interfejsu umożliwiający użytkownikowi wybór godziny i minuty, kiedy wywołane będzie zdarzenie czasowe.

Przykład definiowania zdarzenia czasowego z zastosowaniem elementu **Czas** został zaprezentowany w rozdziale 8.3 Timery.

### Edycja kontrolki

- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Kanał dla godzin** - Nazwa identyfikatora, w którym będzie zapamiętywana godzina wybrana przez użytkownika.
- **Kanał dla minut** - Nazwa identyfikatora, w którym będzie zapamiętywana minuta wybrana przez użytkownika.

Najczęściej dla minut i godzin używane są identyfikatory odnoszące się do zmiennych nieulotnych typu MEM, ponieważ zapewniają, że nastawy wprowadzone przez użytkownika nie zostaną utracone nawet pod wyłączeniu zasilania modułu Base.

## LCN

Grupa kontrolki LCN zawiera elementy dedykowane do zastosowania w połączeniu z modułami LCN.

### 1. Ściemniacz

**Ściemniacz** przeznaczony jest do sterowania wyjściem ściemnianym w module LCN w zadeklarowanym przedziale wartości. Domyślny przedział to 0-100.

**Ściemniacz** jest wyświetlany w aplikacji Remote jako suwak. Kliknięcie na etykiecie z aktualną wartością realizuje funkcję przełączającą pomiędzy minimalną i maksymalną wartością przypisanego zakresu.

### Edycja kontrolki

- **Segment** - Numer segmentu w systemie LCN, w którym znajduje się wybrany moduł LCN.
- **Moduł** - ID modułu LCN w obrębie wybranego segmentu.
- **Wybierz z listy** - Po kliknięciu wyświetlone zostanie okno z listą modułów LCN dostępnych w instalacji, umożliwiające wybór adresata komendy.
- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Wyjście** - Numer wyjścia ściemnianego (1-4).
- **Minimum** - Minimalna wartość na jaką można ustawić suwak. Zakres od 0 do 100.
- **Maksimum** - Maksymalna wartość na jaką można ustawić suwak. Zakres od 1 do 100.
- **Skok** - Określa minimalną zmianę, jaką można wprowadzić przesuwanym suwakiem. Domyślna wartość: 1. Zakres od 1 do 99.

### 2. Przełącznik

**Przełącznik** umożliwia zdefiniowanie wielofunkcyjnego elementu interfejsu. W zależności od opcji wybranej w polu **Typ**, realizowane są inne funkcje. Inna jest także deklaracja tego elementu dla poszczególnych opcji:

### Przełącznik

Jeżeli wybrany został **Przełącznik**, wówczas kontrolka steruje przełącznikiem w module LCN.

### Edycja kontrolki

- **Segment** - Numer segmentu w systemie LCN, w którym znajduje się wybrany moduł LCN.
- **Moduł** - ID modułu LCN w obrębie wybranego segmentu.
- **Wybierz z listy** - Po kliknięciu wyświetlone zostanie okno z listą modułów LCN dostępnych w instalacji.
- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Typ** - *Przełącznik*
- **Przełącznik**: Numer przełącznika, który jest przyporządkowany do tego elementu. Możliwe wartości: od 1 do 8.
- **Odwróć wartość**: Zaznaczenie tej opcji powoduje odwrócenie logiki działania przycisku.

### Wyjście

Jeżeli jako typ zostało wybrane **Wyjście**, wówczas **Przełącznik** steruje wyjściem ściemnianym modułu LCN, ale w trybie: załącz/wyłącz

### Edycja kontrolki

- **Segment** - Numer segmentu w systemie LCN, w którym znajduje się wybrany moduł LCN.
- **Moduł** - ID modułu LCN w obrębie wybranego segmentu.

## 7. Konfiguracja aplikacji sterujących

- **Wybierz z listy** - Po kliknięciu wyświetlone zostanie okno z listą modułów LCN dostępnych w instalacji.
- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Wyjście**: Numer wyjścia proporcjonalnego, które jest przyporządkowane do tego elementu. Możliwe wartości to **1-4**.
- **Próg**: Wartość wyjścia od której przyjmuje się, że wyjście jest włączone. Dozwolony zakres: od **1-100**
- **Rampa**: Wartość rampy z jaką wyjście zmienia wartość. Dostępny zakres: **0-20**.
- **Odwróć wartość**: Zaznaczenie tej opcji powoduje odwrócenie logiki działania przycisku.

## 3. Temperatura

**Temperatura** to element interfejsu użytkownika, który wyświetla aktualną i zadaną temperaturę. Po kliknięciu na kontrolce wyświetlana jest nowa strona z wykresem zmian temperatury (ostatnie 24h) i możliwością zmiany temperatury zadanej (tylko jeżeli wybrano **Regulator** jako sposób regulacji temperatury).

### Edycja kontrolki

Na definicję tego elementu składa się:

- **Segment** - Numer segmentu w systemie LCN, w którym znajduje się wybrany moduł LCN.
- **Moduł** - ID modułu LCN w obrębie wybranego segmentu.
- **Wybierz z listy** - Po kliknięciu wyświetlone zostanie okno z listą modułów LCN dostępnych w instalacji, umożliwiające wybór adresata komendy.
- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Czujnik**: Zmienna w module LCN z której odczytywana jest aktualna temperatura, możliwe wartości to **TVar**, **R1Var** lub **R2Var** (w nowej generacji modułów odpowiada to zmiennym od 1 do 3).
- **Sposób regulacji**.: Dostępne opcje: **Regulator**, **Wartość progowa**. Jeżeli wybrano **Regulator**, wówczas temperatura będzie sterowana przy użyciu proporcjonalnego regulatora LCN. Jeżeli wybrano **Wartość progową**, sterowanie będzie realizowane przy użyciu wartości progowych zaprogramowanych w LCN-Pro.
- **Minimum**: Minimalna wartość temperatury, jaką można ustawić przy użyciu kontrolki.
- **Maksimum**: Maksymalna wartość temperatury, jaką można ustawić przy użyciu kontrolki.

**Regulator:**

- **Regulator**: Regulator w którym ustawiana jest zadana temperatura. Dopuszczalne wartości: **1-2**.

**Wartość progowa**

- **Wartość progowa** - numer wartości progowej, której dotyczy sterowanie.
- **Wartość zaprogramowana** - Zaprogramowana wartość progowa, względem której są zmiany relatywne. Zakres: **-200:200**

## 4. Roleta

**Roleta** to elementu interfejsu, który steruje silnikiem do rolet.

### Edycja kontrolki

- **Segment** - Numer segmentu w systemie LCN, w którym znajduje się wybrany moduł LCN.
- **Moduł** - ID modułu LCN w obrębie wybranego segmentu.
- **Wybierz z listy** - Po kliknięciu wyświetlone zostanie okno z listą modułów LCN dostępnych w instalacji, umożliwiające wybór adresata komendy.
- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Typ** - Sposób sterowania roletami. Dostępne są następujące typy:

- Przekazniki (bez pozycjonowania) - roleta sterowana z użyciem pary przekazników, bez informacji o pozycjonowaniu.
- Przekazniki (z pozycjonowaniem) - roleta sterowana z użyciem pary przekazników z informacją o pozycjonowaniu (wymagany moduł LCN-BS4).
- Wyjście analogowe - rolety podłączone do wyjść ściemnianych w module LCN.
- **Silnik:** Numer silnika do którego podłączona jest roleta. Dostępny tylko w przypadku wyboru sterowania rolet za pomocą przekazników. Każdy silnik wymaga pary przekazników. Pierwszy przekaznik z pary służy do zasilania silnika, natomiast drugi odpowiada za kierunek ruchu rolety.  
Do wyboru:
  - **1,2,3,4** - sterowanie pojedynczymi silnikami
  - **1+2, 3+4** - sterowanie parami rolet
  - **1-4** - sterowanie całym blokiem przekazników - 4 rolety

### 5. Wejście binarne LCN

**Wejście binarne LCN** wyświetla aktualny stan wejścia binarnego w module LCN.

#### Edycja kontrolki

- **Segment** - Numer segmentu w systemie LCN, w którym znajduje się wybrany moduł LCN.
- **Moduł** - ID modułu LCN w obrębie wybranego segmentu.
- **Wybierz z listy** - Po kliknięciu wyświetlone zostanie okno z listą modułów LCN dostępnych w instalacji, umożliwiające wybór adresata komendy.
- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Czunik:** Numer wejścia binarnego LCN, którego stan ma być pokazywany. Dozwolona wartość od **1-8**
- **Ikona wł.:** Sposób reprezentowania wejścia w stanie włączony.
- **Ikona wył.:** Sposób reprezentowania wejścia w stanie wyłączony.

## 7. Konfiguracja aplikacji sterujących

### IDS

Elementy z grupy **IDS** (Intrusion Detection System - System Wykrywania Włamań) są dedykowane do zastosowania z systemem alarmowym SATEL Integra.

#### 1. Wejście IDS

**Wejście IDS** to kontrolka, która wyświetla aktualny stan wejścia systemu alarmowego. Kontrolka może być użyta np. do wyświetlania stanu otwarcia okien, drzwi, czujników PIR.

#### Edycja kontrolki

- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Wejście** - Numer wejścia w systemie IDS, którego stan ma być wyświetlany. Dozwolona wartość zależy od ilości wejść w danej centrali alarmowej (max. 256).
- **Ikona wł.:** Sposób reprezentowania wejścia w stanie włączonym.
- **Ikona wył.:** Sposób reprezentowania wejścia w stanie wyłączonym.

#### 2. Wyjście IDS

**Wyjście IDS** to kontrolka, która wyświetla aktualny stan pojedynczego wyjścia systemu alarmowego.

#### Edycja kontrolki

- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Wyjście** - Numer wyjścia w systemie IDS, którego stan ma być wyświetlany. Dozwolona wartość zależy od ilości wyjść w danej centrali alarmowej (max. 256).
- **Ikona wł.:** Sposób reprezentowania wyjścia w stanie włączonym.
- **Ikona wył.:** Sposób reprezentowania wyjścia w stanie wyłączonym.

#### 3. Strefa IDS

**Strefa IDS** to element interfejsu użytkownika, który wyświetla aktualny stan strefy alarmowej oraz umożliwia jej uzbrajanie i rozbrajanie. Po kliknięciu na kontrolce wyświetlana jest ekran z klawiaturą do wprowadzenia kodu pin.

Stan strefy alarmowej reprezentowany jest przez zróżnicowany wygląd kontrolki. Wyróżnia się pięć różnych stanów:

1. **Kontrolka szara** - strefa rozbrojona.
2. **Kontrolka zielona** - strefa uzbrojona.
3. **Kontrolka czerwona** - alarm.
4. **Miganie szary/zielony** - czas na wyjście.
5. **Miganie szary/czerwony** - czas na wejście.

#### Edycja kontrolki

- **Etykieta** - Etykieta wyświetlana w menu aplikacji Remote.
- **Strefa:** Numer strefy alarmowej.



## 7.2. Wizualizacja

Zakładka **Wizualizacja** zawiera edytor do tworzenia wizualizacji wyświetlanej w aplikacjach sterujących DOMIQ.

Tryb wizualizacji jest znakiem rozpoznawczym systemu DOMIQ. Jako pierwsi zaimplementowaliśmy ten rodzaj interfejsu w aplikacji mobilnej do sterowania automatyką budynkową.

Każda wizualizacja składa się z tła - najczęściej jest to grafika przedstawiająca rzut kondygnacji oraz kontrolki, które mogą sterować urządzeniami i wyświetlać ich bieżący stan. Wizualizacja może składać się z wielu ekranów, dzięki czemu możliwa jest stworzenie nawet bardzo złożonych interfejsów użytkownika.

Jest to bardzo wygodny w obsłudze sposób sterowania, gdyż użytkownik na jednym ekranie ma dostęp do wszystkich najczęściej używanych funkcji systemu. Tworzenie wizualizacji jest bardzo proste i nie wymaga żadnej wiedzy programistycznej. Sterowanie domem przy użyciu wizualizacji może odbywać się na trzy sposoby:

- Przy użyciu aplikacji DOMIQ/Remote - po obrocie urządzenia do pozycji poziomej.
- Z użyciem aplikacji DOMIQ/AirDisplay.
- Z poziomu dowolnego komputera przez przeglądarkę internetową. W tym celu należy wpisać adres IP modułu w polu adresu w przeglądarce internetowej. Następnie zalogować się z użyciem przycisku **Wizualizacja**. W przypadku tej metody, bieżący stan kontrolki będzie widoczny wyłącznie, gdy komputer podłączony jest do tej samej sieci lokalnej, w której znajduje się moduł **Base**. W przypadku połączeń zdalnych należy skorzystać z tunelu VPN. W ostateczności można przekierować port TCP 4225. **Jest to jednak rozwiązanie przez nas stanowczo niezalecane, gdyż w ten sposób osoby nieupowaznione mogą przejąć kontrolę nad instalacją - port 4225 nie posiada szyfrowania, a umożliwia sterowanie wszystkimi urządzeniami w instalacji.**

### Pasek przycisków

W górnej części okna znajduje się pasek z przyciskami.

- **+ Ekran** - Kliknij, aby dodać nowy ekran do struktury wizualizacji.
- **+ Warstwa** - Kliknij, aby dodać nową warstwę do ekranu.
- **Kopiuuj** - Kopiuje do schowka aktualnie zaznaczony element wizualizacji (może być kontrolka lub element struktury).
- **Wklej** - Wkleja skopiowany element wizualizacji.
- **Usuń** - Kliknij, aby usunąć zaznaczone elementy wizualizacji.
- **Importuj** - Umożliwia zaimportowanie pliku .xml z konfiguracją.
- **Eksportuj** - Umożliwia zapis do pliku .xml bieżącej struktury interfejsu.
- **Eksportuj zaznaczone** - Umożliwia zapis do pliku jedynie tych elementów, które mają zaznaczone pole wyboru. Często sytuacją jest powielanie danej funkcjonalności w kolejnych instalacjach. W ten sposób możesz wyeksportować tylko część interfejsu i użyć jej w innej lokalizacji.
- **Zapisz** - Zapisuje bieżącą konfigurację w pamięci modułu Base.
- **Przywróć** - Wczytuje do drzewa struktury ostatnio zapisaną konfigurację.

### Tabela atrybutów

W dolnej lewej części okna znajduje się **Tabela atrybutów**, która zawiera informacja o zaznaczonym elemencie wizualizacji wraz z opcjami edycyjnymi tego elementu.

### Struktura

W oknie **Struktury** wyświetlane jest drzewo struktury wizualizacji. Drzewo ma budowę hierarchiczną, tak jak w przypadku zakładki **Remote**.

- Nowe elementy dodawane są na dwa sposoby:

## 7. Konfiguracja aplikacji sterujących

- z użyciem przycisków - tylko **Ekran** i **Warstwa**.
- metodą **Przeciągnij i upuść** - wszystkie kontrolki. O sposobie dodawania kontrolki przeczytasz w dalszej części tego rozdziału.
- Aby zmienić pozycję elementu w strukturze, przeciągnij wybrany element i upuść w nowym miejscu.
- Jeżeli element nie może być dodany w wybranym miejscu, wówczas zobaczysz czerwony znacznik. Dla prawidłowych operacji znacznik ma kolor zielony.
- W drzewie struktury obowiązują kilka zasad dotyczących hierarchii elementów:
  - Na głównym poziomie można dodać tylko elementy typu **Ekran**.
  - **Ekran z nadanym identyfikatorem *main* będzie wyświetlany jako domyślny w aplikacji sterującej.**
  - Jeżeli żaden ekran nie ma identyfikatora *main*, wówczas domyślnie będzie wyświetlony pierwszy ekran z listy.
  - Bezpośrednio wewnątrz **Ekranów** mogą znajdować się wszystkie dostępne elementy wizualizacji, z wyjątkiem innych elementów typu **Ekran**.
  - Wewnątrz **Warstw** mogą znajdować się wszystkie dostępne kontrolki, z wyjątkiem innej **Warstwy** i **Ekranów**.
- Aby zmienić właściwości wybranego elementu, należy kliknąć na nim - jego atrybuty zostaną wyświetlone w **Tabeli atrybutów**.
- Aby usunąć element(y), należy zaznaczyć go przez kliknięcie w pole wyboru po prawej stronie etykiety, a następnie kliknąć przycisk **Usuń** lub klawisz **backspace** na klawiaturze. Jeśli usuwany element interfejsu ma zagnieżdżone elementy, wówczas one również zostaną usunięte.
- Aby skopiować element, kliknij na nim (po prawej stronie etykiety pojawi się znacznik błękitny prostokąt), a następnie kliknij przycisk **Kopiuj** lub kombinację klawiszy **ctrl + c**. Element został skopiowany i znajduje się w schowku.
- Aby wkleić skopiowany element użyj przycisku **Wklej** lub skorzystaj ze skrótu klawiaturowego **ctrl + v**. Skopiowana kontrolka zostanie umieszczona w następującej lokalizacji:
  - poniżej aktualnie wybranej kontrolki;
  - na końcu listy, jeżeli zaznaczony jest element **Ekran** lub **Warstwa**;
- Pole wyszukiwania pozwala na szybkie wyszukanie elementów w strukturze. Możliwe jest przeszukiwanie na podstawie etykiety oraz typu kontrolki. Wpisując w pasku wyszukiwania: *światło*, zostaną wyświetlone wszystkie kontrolki tego typu, występujące w całej konfiguracji.
- Pole wyboru **Zaznacz wszystko** umożliwia szybkie zaznaczenie wszystkich elementów struktury. Jest to szczególnie przydatne, kiedy chcesz szybko usunąć całą strukturę.

### 1. Ekran

**Ekran** to podstawowy element struktury wizualizacji, który odpowiada pojedynczemu widokowi wyświetlanemu w aplikacji. Aby dodać nowy ekran należy kliknąć **+Ekran**. Nowy ekran jest zawsze dodawany na końcu drzewa.

**WAŻNE! Każdy ekran musi mieć unikalny identyfikator w strukturze wizualizacji. Po dodaniu, ekran ma nadany losowy identyfikator, który zalecamy zmienić na bardziej czytelny dla użytkownika.**

### Edycja kontrolki

- **Identyfikator** - Unikalny identyfikator ekranu. Identyfikator służy m.in do nawigowania pomiędzy ekranami. .
- **Obrazek tła** - Tu możesz wybrać grafikę znajdującą się w pamięci modułu Base, która będzie wyświetlana jako tło ekranu. Obsługiwane są obrazki o rozmiarze maksymalnie 250kB. Zalecana rozdzielczość 800x600 px
- **URL** - Adres URL do grafiki hostowanej na zewnętrznych serwerach. Jest to szczególnie przydatne, gdy chcesz użyć grafik o większych rozdzielczościach i rozmiarach pliku.

- **Zablokowany** - Ustawienie blokady dla ekranu oznacza, że zablokowane zostaje przemieszczanie elementów wizualizacji w oknie **Edytor wizualizacji**. Możliwa jest wciąż edycja ich parametrów. Blokadę można również włączyć/wyłączyć klikając na ikonkę kłódki znajdującej się po prawej stronie identyfikatora ekranu. Włączenie blokady ekranu nie blokuje warstw dodanych do ekranu.
- **Siatka** - Parametr **siatka** tworzy niewidzialną siatkę, względem której układane są elementy wizualizacji. Wpisanie w polu **siatka** liczby 5 oznacza, że minimalne przemieszczenie elementu na ekranie wizualizacji to 5 pikseli.
- **Kolor** - Umożliwia wybór koloru tła ekranu, w przypadku gdy nie jest użyte tło graficzne.

## Zmiana wyświetlanego ekranu

Aktualnie wyświetlany ekran może być zmieniony na dwa sposoby:

1. Za pomocą funkcji nawigacyjnych kontrolki **Przycisk**, co zostało opisane w dalszej części tego rozdziału.
2. Za pomocą komend. Komendy są obsługiwane tylko w aplikacji DOMIQ/Remote. Jest to szczególnie przydatne, gdy tablet jest zamontowany na ścianie jako panel sterowania. Ekran wizualizacji może zostać dynamicznie zmieniony na podstawie zdarzenia, które wystąpiło w systemie. Np. wyświetlenie ekranu z podglądem z kamery w wyniku naciśnięcia przycisku na domofonie. Innym scenariuszem może być wyświetlenie ekranu do zarządzania systemem alarmowym w wyniku zaistnienia zdarzenia alarmowego. Komenda `DISPLAY . screen` służąca do zmiany ekranu została opisana w rozdziale 12. Identyfikatory

## 2. Warstwa

**Warstwa** może być użyta wyłącznie jako podelement **strony**. Warstwy pozwalają na wygodne grupowanie elementów tego samego rodzaju np. włączniki świateł, wskazania temperatur, itd. Warstwy są szczególnie przydatne tam, gdzie na wizualizacji jest wyświetlanych jednocześnie dużo kontrolki, gdyż mają one funkcję przełączania widoczności. Dzięki temu można okresowo wyłączyć wyświetlanie części elementów dla poprawienia czytelności wizualizacji.

Aby dodać nową warstwę należy wybrać ekran, a następnie kliknąć przycisk **+Warstwa**. Tak, jak w przypadku elementu **Ekran**, również **Warstwa** musi mieć swój unikalny identyfikator. Każda dodana warstwa ma nadany losowy identyfikator, który zalecamy zmienić na bardziej przyjazny dla użytkownika.

## Edycja kontrolki

- **Identyfikator** - Unikalny identyfikator warstwy. Identyfikator służy do sterowania widocznością warstwy.
- **Widoczny** - Opcja umożliwiająca sterowanie widocznością warstwy z poziomu **Tabeli atrybutów**. Widoczność można włączyć/wyłączyć klikając na ikonkę "oko" znajdującej się po prawej stronie identyfikatora ekranu.
- **Zablokowany** - Ustawienie blokady dla warstwy oznacza, że zablokowane zostaje przemieszczanie elementów wizualizacji w oknie **Edytor wizualizacji**. Możliwa jest wciąż edycja ich parametrów. Blokadę można również włączyć/wyłączyć klikając na ikonkę kłódki znajdującej się po prawej stronie identyfikatora ekranu. Włączenie blokady ekranu nie blokuje warstw dodanych do ekranu.

Kolejność warstw w strukturze może być swobodnie zmieniana, identycznie jak w przypadku ekranów. Jednakże kolejność ma wpływ na wyświetlanie warstw na wizualizacji wg zasady: **im niżej warstwa w strukturze, tym bardziej na wierzchu jest wyświetlana w aplikacji.**

## Przełączanie widoczności warstwy

Widoczność warstwy można zmieniać na dwa sposoby:

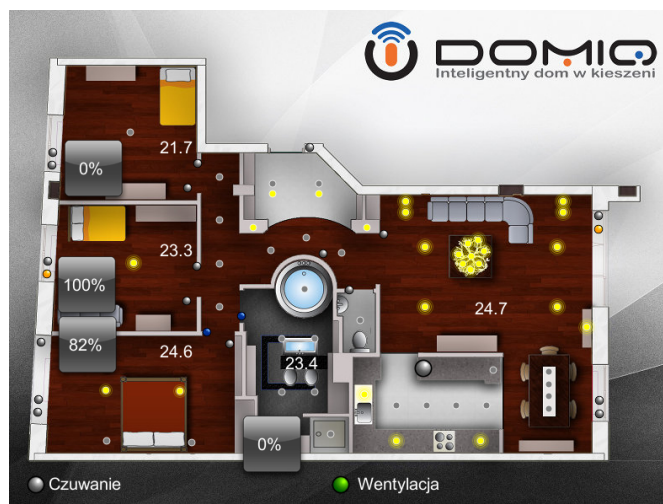
1. Za pomocą funkcji nawigacyjnych kontrolki **Przycisk**, co zostało opisane w dalszej części tego rozdziału.

## 7. Konfiguracja aplikacji sterujących

2. Za pomocą komend. Komendy są obsługiwane tylko w aplikacji **DOMIQ/Remote**. Komenda `DI-SPLAY.layer` służąca do przełączania widoczności warstwy została opisana w rozdziale 12. Identyfikatory.

## Edytor wizualizacji

**Edytor wizualizacji** to graficzna część zakładki **Wizualizacja**, w której tworzony jest wygląd wizualizacji poprzez dodanie kontroltek, ustawienie ich atrybutów oraz rozmieszczenie.



Rys 7.1: Przykładowa wizualizacja

## Dodawanie/usuwanie kontroltek

W górnej części edytora znajduje się pasek dostępnych kontroltek. Nowe elementy są dodawane metodą „przeciągnij i upuść”. Dodany element automatycznie pojawi się również w drzewie **Struktura**.

Aby skopiować element, zaznacz go i kliknij **Kopiuj** lub użyj skrótu klawiaturowego **ctrl + c**.

Aby wkleić skopiowany element, kliknij przycisk **Wklej** lub użyj skrótu klawiaturowego **ctrl + v**.

Aby skasować element, zaznacz go i kliknij **Usuń** lub naciśnij klawisz **Backspace** na klawiaturze.

Powyższe trzy reguły można stosować również, gdy zaznaczonych jest wiele kontroltek.

## Rozmieszczenie kontroltek

Do zmiany położenia kontrolki możesz użyć jedną z dostępnych metod. Zaznacz kontrolkę klikając na niej, a następnie:

- Trzymając wciśnięty lewy przycisk myszki zmień położenie kontrolki.
- Używając klawiszy strzałek na klawiaturze. Każde naciśnięcie klawisza zmienia położenie o 1 piksel w danym kierunku. Robiąc to samo, ale z wciśniętym jednocześnie klawiszem **Shift** zmienia położenie o 10 pikseli.
- Zmień atrybuty **x** oraz **y** w **Tabeli atrybutów**. Punkt o współrzędnych 0,0 odpowiada lewemu górnemu narożnikowi okna wizualizacji.

Powyższe sposoby (poza edycją atrybutów **x** i **y**) można również zastosować do jednoczesnej zmiany położenia wielu kontroltek. Aby zaznaczyć kilka kontroltek, klikaj na nich mając jednocześnie wciśnięty na klawiaturze klawisz **ctrl** lub **command** (Mac).

Ponadto powyżej okna wizualizacji znajdują się przyciski ułatwiające rozmieszczenia wielu kontroltek kontroltek:

- **Wyrównaj do lewej** - ustawia wszystkie kontrolki w linii wyrównując je do lewej. Kontrolki zostaną wyrównane do kontrolki wysuniętej najbardziej na lewo spośród zaznaczonych.
- **Wyrównaj do prawej** - ustawia wszystkie kontrolki w linii wyrównując je do prawej. Kontrolki zostaną wyrównane do kontrolki wysuniętej najbardziej na prawo spośród zaznaczonych.
- **Wyrównaj do góry** - ustawia wszystkie kontrolki w linii wyrównując je do góry. Kontrolki zostaną wyrównane do kontrolki wysuniętej najbardziej w górę spośród zaznaczonych.
- **Wyrównaj do dołu** - ustawia wszystkie kontrolki w linii wyrównując je do dołu. Kontrolki zostaną wyrównane do kontrolki wysuniętej najbardziej w dół spośród zaznaczonych.

## 7. Konfiguracja aplikacji sterujących

- **Wyrównaj odstępy pionowe** - ustawia równomierne odstępy pionowe pomiędzy zaznaczonymi kontrolkami.
- **Wyrównaj odstępy poziome** - ustawia równomierne odstępy poziome pomiędzy zaznaczonymi kontrolkami.

Aby odznaczyć wybrane kontrolki, kliknij na tło wizualizacji.

Pole wyboru **Tryb edycji** przełącza pomiędzy trybem edycji wizualizacji a trybem testowania. W trybie testowania, zmiana konfiguracji jest niemożliwa, natomiast okno wizualizacji symuluje działanie wizualizacji - klikanie na kontrolkach powoduje wykonanie komend do nich przypisanych.

W dalszej części tego podrozdziału opisane zostały wszystkie dostępne kontrolki.

### 1. Światło

To najbardziej uniwersalna kontrolka. Może być użyta do sterowania oświetleniem, roletami, strefami alarmowymi oraz do wizualizowania stanu dowolnego identyfikatora np. stanu czujek alarmowych, czujek ruchu, itd.

### Edycja kontrolki

- **x i y** - Współrzędne kontrolki. Wartości **x** i **y** można zmieniać poprzez wpisanie tych wartości z klawiatury lub po prostu przesuwając element w oknie wizualizacji trzymając wciśnięty lewy przycisk myszy.
- **Szerokość i Wysokość** (tylko do odczytu) - Rozmiary kontrolki w pikselach.
- **Warstwa** (tylko do odczytu) - Warstwa, na której znajduje się kontrolka.
- **Kanał** - Identyfikator, którym ten element będzie sterować oraz którego stan będzie wyświetlany na wizualizacji. Jeżeli równocześnie uzupełnione są pola **Włącz** i **Wyłącz**, wówczas wprowadzony tu identyfikator będzie używany wyłącznie do wyświetlania bieżącego stanu.



- `LCN.relay.0.121.1`  
Sterowanie przekaźnikiem nr 1 w module LCN o adresie 121 oraz wyświetlenie jego aktualnego stanu.
- `LCN.output.0.113.1`  
Sterowanie wyjściem ściemnianym nr 1 w module LCN o adresie 113 oraz wyświetlenie jego aktualnego stanu.

- **Motyw** - Zestaw ikon używanych do graficznej reprezentacji kontrolki.
- **Sposób kontroli** - Określa tryb pracy kontrolki.
  - **Brak**: Kontrolka będzie wyświetlać wyłącznie stan identyfikatora, ale niemożliwe będzie sterowanie.
  - **Ściemniacz**: Ta opcja może być stosowana wyłącznie z wyjściami ściemnianymi. Pozwala na płynną regulację jasności światła - w tym celu dotknij i przytrzymaj kontrolkę. Ponowne długie naciśnięcie rozpoczyna płynną zmianę jasności, ale w przeciwnym kierunku. Kiedy puścisz, zmiana jasności zatrzyma się. Krótkie naciśnięcie wysyła komendę włącz/wyłącz.
  - **Wł./Wył.**: Tradycyjny włącznik dwustanowy. Każde naciśnięcie wysyła na przemian komendy włącz/wyłącz.
- **Parametr** - Pole do wprowadzenia dodatkowych parametrów, które będą wysłane wraz z komendą.



- `ramp: 3` - Do komendy zostanie dołączony parametr `ramp` z wartością 3.
- `pin: 12345` - Do komendy zostanie dołączony parametr `pin` z wartością 12345.

- **Włącz** - Komenda, które zostanie wysłana po naciśnięciu elementu światło, gdy aktualnie jest on w stanie wyłączonym.





Jeżeli w polu **Włącz** wpiszesz `LCN.output.0.10.1=80`, wówczas po naciśnięciu kontrolki wysterowane zostanie wyjście ściemniane nr 1 do wartości 80% w module LCN o adresie 10.

Jeżeli pole **Włącz** pozostanie puste, naciśnięcie kontrolki powoduje wysyłanie komendy `on` do identyfikatora wpisanego w komórce **Kanał**.

- **Wyłącz** - Polecenie, które zostanie wykonane po naciśnięciu elementu światło, gdy jest aktualnie włączony.



Jeżeli w polu **Wyłącz** wpiszesz `LCN.output.0.10.1=0`, wówczas po naciśnięciu kontrolki wysterowane zostanie wyjście ściemniane nr 1 do wartości 0% w module LCN o adresie 10.

Jeżeli pole **Wyłącz** pozostanie puste, naciśnięcie kontrolki powoduje wysyłanie komendy `off` do identyfikatora wpisanego w komórce **Kanał**.

**UWAGA!** Przy użyciu właściwości **Włącz** i **Wyłącz** ignorowane jest pole **Parametr**. Aby do komendy przekazać dodatkowe atrybuty, należy je zawrzeć w komendzie. Ogólna składnia jest następująca:  
`<identyfikator>;<atrybut_1>:<wartość_1>,<atrybut_n>:<wartość_n>`



- **Włącz**: `LCN.output.0.10.1=100;ramp:5`  
Włącz na 100% wyjście nr 1 w module LCN o adresie 10 z rampą równą 5.
- **Wyłącz**: `LCN.output.0.10.1=0;ramp:5`  
Wyłącz wyjście nr 1 w module LCN o adresie 10 z rampą równą 5.

- **Kod** - Opcja umożliwiająca dodanie kodu, którym chronione będzie wykonywanie komend przypisanych do tej kontrolki. Przy zapisaniu konfiguracji, wprowadzona tu wartość będzie zapisana w formie zaszyfrowanej i również w tej formie będzie wyświetlona następnie w polu **Kod**.

## Przykłady



1. Sterowanie wyjściem ściemnianym LCN z rampą 3

**Kanał:** `LCN.output.0.10.1`

**Sposób kontroli:** *Ściemniacz*

**Parametr:** `ramp:3`

2. Uzbrajanie/rozbrajanie strefy alarmowej **Satel** nr 1.

**Kanał:** `IDS.armed.1`

**Sposób kontroli:** *Przełącznik*

**Parametr:** `pin:12345`

Kod pin należy zastąpić właściwym pinem dla strefy alarmowej.

3. Sterowanie roletą na przekaźnikach LCN.

W tym celu niezbędne są dwie kontrolki typu **Światło**. Po jednej dla ruchu w górę i w dół. Zakładamy, że sterujemy roletą nr 1 w module o adresie 10.

**Światło** nr 1 - ruch dół/stop:

**Kanał:** `LCN.relay.0.10.1` - pierwszy przekaźnik z pary odpowiada za zasilanie rolety.

**Sposób kontroli:** *Przełącznik*

**Motyw** - Motyw musi składać się z dwóch ikon: jednej wyświetlającej strzałkę z kierunkiem ruchu, a drugi zawierając połowę napisu STOP, który będzie wyświetlany, gdy roleta jest w ruchu.

**Włącz:** `LCN.relays.0.10=11-----`

**Wyłącz:** `LCN.relays.0.10=0-----`

**Światło** nr 2 - ruch góra/stop:

**Motyw:** Motyw przedstawiający ruch góra/stop

**Włącz:** `LCN.relays.0.10=10-----`

Pozostałe parametry są identyczne, jak dla pierwszej kontrolki.

## 7. Konfiguracja aplikacji sterujących

### 2. Przycisk

**Przycisk** jest kontrolką bezstanową - tzn. że nie wyświetla ona bieżącego stanu identyfikatora. Każde naciśnięcie przycisku realizuje za każdym razem to samo przypisane polecenie.

Ponadto przycisk może realizować funkcję nawigacyjną w strukturze wizualizacji: wywoływać inny ekran lub przełączać widoczność warstwy.

### Edycja kontrolki

- **x i y** - Współrzędne kontrolki. Wartości **x** i **y** można zmieniać poprzez wpisanie tych wartości z klawiatury lub po prostu przesuając element w oknie wizualizacji trzymając wciśnięty lewy przycisk myszy.
- **Szerokość i Wysokość** (tylko do odczytu) - Rozmiary kontrolki w pikselach.
- **Warstwa** (tylko do odczytu) - Warstwa, na której znajduje się kontrolka.
- **Etykieta** - Wprowadzona tu wartość będzie wyświetlana jako etykieta przycisku.
- **Ekran** - Zawiera listę dostępnych ekranów. Jeżeli z listy zostanie wybrany ekran, wówczas naciśnięcie przycisku przełączy widok do wybranego ekranu.
- **Warstwa** - Zawiera listę dostępnych warstw. Jeżeli z listy zostanie wybrana warstwa, wówczas naciśnięcie przycisku steruje widocznością wybranej warstwy.
- **Komenda** - Komenda, która ma być wykonywana przy każdym naciśnięciu przycisku. Komenda będzie wykonywana także, jeżeli do przycisku została przypisana funkcja nawigacyjna (zmiana ekranu lub przełączanie widoczności warstwy).
- **Motyw** - Zestaw ikon używanych do graficznej reprezentacji przycisku.
- **Kolor** - Kolor etykiety przycisku.
- **Rozmiar** - Rozmiar czcionki na etykiecie przycisku.
- **Kod** - Opcja umożliwiająca wprowadzenie kodu, którym chronione będzie wykonywanie komend przypisanych do tej kontrolki. Przy zapisaniu konfiguracji, wprowadzona tu wartość będzie zapisana w formie zaszyfrowanej i również w tej formie będzie wyświetlona w polu **Kod**. Kod może zabezpieczać zarówno funkcje nawigacyjne (ochrona przed nieautoryzowanym dostępem do wybranych ekranów/warstw wizualizacji), jak i przed wywołaniem komend wprowadzonych w polu **Komenda**.

### 3. Przełącznik

Jest to przycisk dwustanowy. Może być wykorzystany do sterowania dowolnym identyfikatorem na zasadzie włącz/wyłącz np. do włączania i wyłączania oświetlenia, wentylacji itp. Aktualny stan sterowanego urządzenia wyświetlany jest na przycisku.

### Edycja kontrolki

- **x i y** - Współrzędne kontrolki. Wartości **x** i **y** można zmieniać poprzez wpisanie tych wartości z klawiatury lub po prostu przesuując element w oknie wizualizacji trzymając wciśnięty lewy przycisk myszy.
- **Szerokość i Wysokość** (tylko do odczytu) - Rozmiary kontrolki w pikselach
- **Warstwa** (tylko do odczytu) - Warstwa, na której znajduje się kontrolka
- **Kanał** - Identyfikator, którym ten element będzie sterować oraz którego stan będzie wyświetlany na ekranie wizualizacji. Jeżeli równocześnie uzupełnione są pola **Włącz** i **Wyłącz**, wówczas wprowadzony tu identyfikator będzie używany wyłącznie do wyświetlania bieżącego stanu.



- `LCN.relay.0.121.1`  
sterowanie przekaźnikiem nr 1 w urządzeniu o adresie 121 oraz wyświetlanie jego aktualnego stanu.
- `LCN.output.0.113.1`  
sterowanie wyjściem ściemnianym nr 1 w urządzeniu o adresie 113 oraz wyświetlanie jego aktualnego stanu.



- **Parametr** - Pole do wprowadzenia dodatkowych parametrów, które będą wysłane wraz z komendą.
- **Włącz i Wyłącz** - patrz opis tych parametrów w kontrolce **Światło**.
- **Motyw** - Graficzna reprezentacja przełącznika.
- **Kolor** - Kolor etykiety przycisku.
- **Rozmiar** - Rozmiar czcionki etykiety przycisku.
- **Kod** - patrz opis tego parametru w kontrolce **Światło**.

#### 4. Ściemniacz

Kontrolka do sterowania wyjściami ściemnianymi. Krótkie naciśnięcie wysyła komendę włącz/wyłącz. Długie naciśnięcie (powyżej sekundy) rozpoczyna płynną zmianę jasności. Kiedy puścisz przycisk, zmiana jasności zatrzyma się. Ponowne długie naciśnięcie rozpoczyna płynną zmianę jasności, ale w przeciwnym kierunku.

Na przycisku wyświetlana jest procentowo aktualna wartość wyjścia ściemnianego. Ściemniacz posiada identyczny zestaw właściwości jak przycisk **Przełącznik**.

#### 5. Wartość

Kontrolka **Wartość** to uniwersalny element interfejsu do wyświetlania wartości dowolnego identyfikatora.

#### Edycja kontrolki

- **x i y** - Współrzędne kontrolki. Wartości **x** i **y** można zmieniać poprzez wpisanie tych wartości z klawiatury lub po prostu przesuając element w oknie wizualizacji trzymając wciśnięty lewy przycisk myszy.
- **Szerokość i Wysokość** (tylko do odczytu) - Rozmiary kontrolki w pikselach.
- **Warstwa** (tylko do odczytu) - Warstwa, na której znajduje się kontrolka.
- **Kanał** - Identyfikator, którym ten element będzie sterować oraz którego stan będzie wyświetlany na ekranie wizualizacji.
- **Przedrostek** - Tekst, który będzie wyświetlany przed wartością.
- **Przyrostek** - Tekst, który będzie wyświetlany za wartością, np. jednostka miary.
- **Mnożnik** - Wartość przez którą pomnożona będzie aktualna wartość identyfikatora.
- **Przesunięcie** - Wprowadzona tu wartość będzie dodana do aktualnej wartości identyfikatora.
- **Kolor** - Kolor czcionki.
- **Format** - Opcja służąca do formatowania wyświetlanej wartości.

#### Przykład formatowania

Założmy, że bieżąca wartość identyfikatora to 25,43.	
FORMAT	WYNIK
.000	25.430°C
.0##	25.43°C
0.##	25.43°C
000.000	025.430°C



**Podsumowując:** 0 w formatowaniu oznacza, że cyfra na danej pozycji w liczbie zawsze zostanie wyświetlona. Jeżeli nie ma wartości, wówczas zostanie uzupełniona zerami.

## 7. Konfiguracja aplikacji sterujących

Znak # oznacza, że cyfra na danej pozycji będzie wyświetlona jedynie wtedy, gdy jej wartość jest niezerowa. Znak # może być umieszczany jedynie po kropce dziesiętnej.

Jeżeli pole **Format** pozostanie puste, wówczas wartość temperatury będzie wyświetlona bez formatowania.

- **Rozmiar** - Rozmiar czcionki.
- **Minimum** - Minimalna wartość, którą można ustawić.
- **Maksimum** - Maksymalna wartość, którą można ustawić.

Jeśli pola **Minimum** i **Maksimum** zostaną uzupełnione, wówczas kontrolka może służyć także do wprowadzania wartości - po kliknięciu na kontrolce wyświetlona zostanie klawiatura numeryczna. Wartości **Minimum** i **Maksimum** stanowią wówczas zakres dopuszczalnych wartości. Dzięki temu kontrolka **Wartość** być użyta do nastawiania i wyświetlania temperatury, zmian wartości zmiennych VAR/MEM, itd.

## 6. Temperatura

Element dedykowany do wyświetlania i sterowania temperaturą w systemie LCN. Kontrolka posiada ten sam zestaw parametrów, jak element **Wartość**.

## 7. Tekst

Element typu **Tekst** pozwala na wyświetlanie dowolnych treści tekstowych.

### Edycja kontrolki

- **x i y** - Współrzędne kontrolki. Wartości **x** i **y** można zmieniać poprzez wpisanie tych wartości z klawiatury lub po prostu przesuając element w oknie wizualizacji trzymając wciśnięty lewy przycisk myszy.
- **Szerokość i Wysokość** (tylko do odczytu) - Rozmiary kontrolki w pikselach.
- **Warstwa** (tylko do odczytu) - Warstwa, na której znajduje się kontrolka.
- **Tekst** - Wprowadzona tu wartość zostanie wyświetlona na wizualizacji.
- **Kanał** - Nazwa identyfikatora, którego stan chcesz wyświetlić. Jeżeli to pole jest uzupełnione, wówczas ignorowana jest zawartość pola **Tekst**.
- **Kolor** - Kolor tekstu.
- **Rozmiar** - Rozmiar czcionki.

## 8. RGBW

Kontrolka przeznaczona jest do sterowania oświetleniem RGBW. Domyślnie wyświetla bieżący kolor światła oraz etykietę z aktualną jasnością. Po kliknięciu na kontrolce, wyświetlony zostanie widok do nastawiania koloru, jasności i nasycenia - jest to ten sam widok, jaki został opisany w kontrolce **Światło RGBW** w zakładce **Remote**.

### Edycja kontrolki

- **x i y** - Współrzędne kontrolki. Wartości **x** i **y** można zmieniać poprzez wpisanie tych wartości z klawiatury lub po prostu przesuując element w oknie wizualizacji trzymając wciśnięty lewy przycisk myszy.
- **Szerokość i Wysokość** (tylko do odczytu) - Rozmiary kontrolki w pikselach.
- **Warstwa** (tylko do odczytu) - Warstwa, na której znajduje się kontrolka.
- **Czerwony** - Identyfikator, który odpowiada za sterowanie składową czerwoną.
- **Zielony** - Identyfikator, który odpowiada za sterowanie składową zieloną.
- **Niebieski** - Identyfikator, który odpowiada za sterowanie składową niebieską.
- **Biały** - Identyfikator, który odpowiada za sterowanie składową białą.
- **Etykieta** - Etykieta kontrolki wyświetlana w drzewie struktury.
- **Kolor** - Kolor, którym wyświetlana jest bieżąca jasność oświetlenia.

## 9. Kamera

Element **Kamera** służy jest do wyświetlania obrazu z kamer.

### Edycja kontrolki

- **x i y** - Współrzędne kontrolki. Wartości **x** i **y** można zmieniać poprzez wpisanie tych wartości z klawiatury lub po prostu przesuając element w oknie wizualizacji trzymając wciśnięty lewy przycisk myszy.
- **Warstwa** (tylko do odczytu) - Warstwa, na której znajduje się kontrolka.
- **Szerokość i Wysokość** - Rozmiar elementu w pikselach.
- **URL** - Adres strumienia wideo w formacie MJPEG. W tym polu należy wpisać login i hasło dostępu do kamery (jeżeli jest to wymagane).



Przykładowy URL:

`http://username:password@192.168.10.20/video.mjpeg`

## 10. Obrazek

**Obrazek** służy do dodawania grafik do ekranu wizualizacji.

**UWAGA! W ten sposób nie należy dodawać tła wizualizacji!**

### Edycja kontrolki

- **x i y** - Współrzędne kontrolki. Wartości **x** i **y** można zmieniać poprzez wpisanie tych wartości z klawiatury lub po prostu przesuując element w oknie wizualizacji trzymając wciśnięty lewy przycisk myszy.
- **Szerokość i Wysokość** (tylko do odczytu) - Rozmiary kontrolki w pikselach.
- **Warstwa** (tylko do odczytu) - Warstwa, na której znajduje się kontrolka.
- **Źródło** - Lista dostępnych rysunków. Lista wyświetla zawartość folderu **Obrazki** w zakładki **Zasoby**.
- **URL** - Adres URL do grafiki hostowanej na zewnętrznych serwerach. Ten atrybut ma wyższy priorytet niż **Źródło**. Warto jednak wskazać grafikę lokalną (atrybut **Źródło**) na wypadek niepowodzenia przy pobieraniu grafiki z zewnętrznego serwera.

## 7. Konfiguracja aplikacji sterujących

# Rozdział 8

## Automatyzacja

Automatyzacja jest fundamentalną funkcjonalnością i sercem inteligentnego budynku. Jej zadaniem jest uczynienie systemu automatyki bardziej autonomicznym - system samoczynnie będzie pamiętał o wykonywaniu powtarzalnych czynności lub podjęcie decyzję, jak się zachować na podstawie zaprogramowanych w nim reguł.

Ten rozdział podzielony jest na cztery podrozdziały.

1. **Akcje** - Element interfejsu służący do definiowania listy czynności, które mają być wykonane przez moduł Base. Ten element ma zastosowanie w zdarzeniach i timerach, dlatego będzie opisany przed właściwym opisem obu zakładek.
2. **Zdarzenia** - Reguły logiczne, które są uruchamiane, gdy w systemie wywołana zostanie konkretna akcja lub w odpowiedzi na zmianę stanu urządzenia/zmiennej (identyfikatora). Reguły logiczne mogą być skonfigurowane tak, aby były wykonywane tylko w określonych warunkach. Dają pełną dowolność konfiguracji akcji, jakie zostaną wykonane jako reakcja na wystąpienie zdarzenia.
3. **Timery** - Umożliwiają reguł wykonywanych automatycznie o określonej porze (harmonogramy). Dzięki nim bardzo łatwo jest zautomatyzować powtarzalne czynności np. o określonej godzinie, w wybrany dzień tygodnia lub miesiąca.
4. **Logika** - To najbardziej zaawansowana forma automatyzacji. Wymaga znajomości języka skryptowego Lua. Daje pełną swobodę programiście, aby tworzyć własne skrypty automatyzacyjne, które uwzględniają zarówno zdarzenia zachodzące w systemie, stan identyfikatorów, jak i zależności czasowe. Użycie skryptów Lua jest jednak funkcjonalnością bardzo zaawansowaną i zdecydowana większość funkcjonalności w systemie DOMIQ może być skonfigurowana bez użycia skryptów. Skrypty mogą być również używane do integracji z zewnętrznym oprogramowaniem.

# 8.1. Akcje

Okno **Akcje** to uniwersalny element interfejsu konfiguracyjnego służący do definiowania akcji, które mają być wykonane przez moduł Base. **Akcje** znajdziesz w zakładkach **Zdarzenia**, **Timery** oraz **Remote** (kontrolki **Strona** i **Przycisk**).

Okno ma budowę tabeli, która umożliwia zmianę kolejności wierszy. Kolejność w tabeli ma znaczenie - akcje są wykonywane w kolejności z góry na dół.

- **Aktywny** - odznacz, aby deaktywować wybraną akcję. Nieaktywna akcja ma przekreśloną etykietę.
- **Test** - Przycisk umożliwiający testowanie poprawności konfiguracji. Jeżeli system nie napotka problemu, wówczas akcja zostanie wykonana. Akcje zawierające bindowanie identyfikatorów nie są wykonywane.

Aby wyedytować akcję kliknij na jej etykietę w tabeli.

Aby skasować akcję, zaznacz ją, a następnie kliknij **Usuń**.

Dostępne są trzy rodzaje akcji:

- **+ LCN**
- **+ VAR**
- **+ Komenda**

## + LCN

Okno służy do zdefiniowania komendy, która zostanie wysłana na magistralę LCN. Lewa część okna zawiera ustawienia adresata komendy:

- **Opis** (opcjonalnie) - Jeżeli to pole zostanie uzupełnione, wówczas jego zawartość będzie wyświetlana jako etykieta komendy w tabeli **Akcje**.
- **Segment** - Numer segmentu, do którego kierowana jest komenda.
- Pole wyboru **Moduł/Grupa** - Determinuje typ adresata komendy. Jeżeli wybrana jest opcja **Moduł** wówczas komenda zostanie wysłana do określonego modułu LCN. Jego ID należy wpisać w polu **Moduł**. Natomiast w przypadku opcji **Grupa**, komenda jest wysyłana do grupy modułów LCN wskazanej w polu **Grupa**.
- **Wybierz z listy** - Kliknięcie wyświetla listę modułów LCN dostępnych w instalacji umożliwiając wybór modułu, który ma być odbiorcą komendy.
- **Potwierdzenie** - Jeżeli opcja jest zaznaczona, moduł Base odbierze z magistrali LCN potwierdzenie otrzymania komendy przez adresata. Pole jest domyślnie zaznaczone.

Prawa część okna służy do wyboru typu komendy i jej parametrów.

## 1. Wyjście

Komenda sterująca wyjściem ściemnianym.

- **Wyjście** - Wybierz numer wyjścia, którym chcesz sterować.
- **Wartość** - Procentowa wartość, na którą ma być ustawione wyjście.
- **Rampa** - Czyli czas narastania, z jakim wyjście zostanie wysterowane.

## 2. Przekazniki

Komenda sterująca przekaznikami. Zawiera interfejs do wyboru wartości, jaka zostanie wysłana do poszczególnych przekazników.

## 3. Silniki

Sterowanie silnikami rolet za pomocą przekazników. Możliwe sterowanie od 1 do 4 rolet z pozycjonowaniem (opcja z monitorowaniem prądu – wymagany moduł LCN-BS4) lub bez pozycjonowania. Pola wyboru od **1** do **4** oznaczają pary przekazników. **1** odpowiada przekaznikom nr 1 i 2, **2** przekaznikom 3 oraz 4

itd. Pozycje **1+2** oraz **3+4** przeznaczone są do sterowania parami silników. Natomiast opcja **1-4** steruje wszystkimi 8 przekaźnikami.

#### 4. Wyślij klawisze

Wysłanie polecenia symulującego naciśnięcie dowolnego klawisza (klawiszy) z wybranej tablicy (tablic), z dowolną akcją.

#### 5. Zablokuj klawisze

Polecenie umożliwiające zablokowanie wybranych klawiszy z zadanej tablicy.

#### 6. Scena

Polecenie umożliwiające wywołanie lub zapisanie sceny.

- **Operacja:**
  - **Załaduj:** Wczytanie wybranej sceny.
  - **Zapisz:** Zapisanie sceny. W ustawieniach sceny zostaną zapisane nastawy i numery wyjść aktywne w momencie wysłania komendy.
- **Numer sceny:** Numer sceny do wczytania/zapisania. Zakres od 1 do 10.
- **Aktywne wyjścia:** Numer wyjścia (wyjść), które ma zostać wczytane/zapisane w scenie. Dostępne opcje: **Wyjście 1, Wyjście 2, Wyjście 3, Wyjście 4.**
- **Użyj czasu z definicji sceny:** Jeżeli opcja zaznaczona, wówczas wyjścia będą wysterowane z rampą zaprogramowaną w definicji sceny. W przeciwnym przypadku można ustalić własną rampę w polu **Czas**.

#### 7. Rejestr

Polecenie przełączające rejestr scen. Zakres zmian od 0 do 9.

#### + VAR

Komenda do ustawiania wartości w zmiennej wirtualnej typu **VAR**. Zmienne **VAR** są przechowywane w pamięci modułu **Base** do czasu jego restartu.

- **Opis** (opcjonalnie) - Jeżeli to pole zostanie uzupełnione, wówczas jego zawartość będzie wyświetlana jako etykieta komendy w tabeli **Akcje**.
- **Nazwa** - Tu wprowadź nazwę zmiennej. Będzie ona dostępna w stanie modułu Base pod identyfikatorem `VAR.<nazwa>`. Np. jeżeli nazwa to *test*, wówczas w stanie będzie jako `VAR.test`.
- **Wartość** - Wartość jaka zostanie przypisana do zmiennej.
- **Pamiętaj** - Przekształca zmienną typu **VAR** na typ **MEM**. Wartość zmiennych **MEM** jest zapamiętywana nawet po odłączeniu zasilania od modułu Base. Wówczas zmienna jest dostępna w stanie pod identyfikatorem `MEM.<nazwa>`.

#### + Komenda

Umożliwia wysłanie wartości do dowolnego identyfikatora w systemie DOMIQ (także do zmiennych VAR i modułów LCN). Jest to bardzo potężna funkcjonalność w systemie DOMIQ, gdyż można w ten sposób wykonać dowolną komendę lub wywołać zdarzenie.

- **Opis** (opcjonalnie) - Jeżeli to pole zostanie uzupełnione, wówczas jego zawartość będzie wyświetlana jako etykieta komendy w tabeli **Akcje**.
- **Nazwa** - Tu wprowadź nazwę identyfikatora, do którego chcesz wysłać wartość.  
**WAŻNE!** Zwróć uwagę na konieczność zastosowania prefiku!  
Jeżeli użyjesz prefiksu `C.`, wówczas akcja oznacza wykonanie komendy. Np. włączenie przekaźnika.  
Jeżeli użyjesz prefiksu `E.`, wówczas akcja rozgłosi w systemie zdarzenie, na które moduł Base może zareagować zgodnie z zaprogramowanymi zdarzeniami.
- **Binarna** - Zaznacz to pole, jeżeli przesyłasz surowe dane binarne.

Przestuduj poniższe przykłady:

## 8. Automatyzacja



### 1. Przełączenie przekaźnika nr 3 w module LCN z ID 10.

**Nazwa:** *C.LCN.relay.0.10.3*

**Wartość:** *toggle*

### 2. Ruch w górę roletą nr 2, sterowaną przez moduł DOMIQ/Expander i Satel INT-IORS o adresie 1.

**Nazwa:** *C.EXP.shutter.1.2*

**Wartość:** *up*

### 3. Rozgłoszenie zdarzenia na przełączenie ekranu wizualizacji w aplikacji Remote.

Na potrzeby przykładu, założmy że ekran ma nazwę `test`. Kiedy taka akcja zostanie wykonana, wszystkie aplikacje **Remote**, które aktualnie wyświetlają wizualizację pokażą ekran o nazwie `test`.

**Nazwa:** *E.DISPLAY.screen*

**Wartość:** *test*



## 8.2. Zdarzenia

Zakładka **Zdarzenia** przeznaczona jest to definiowania reguł logicznych, które mają być wykonywane przez moduł Base. Można o nich myśleć, jak o rodzaju makr, które definiują reguły zachowania systemu w reakcji na warunki, które zainstniały. W dalszej części rozdziału będziemy je nazywać zdarzeniami. Kluczowe jest dobre zrozumienie koncepcji tu przedstawionych, dlatego jeżeli coś wydaje Ci się niezrozumiałe, poświęć temu więcej czasu, będzie to czas dobrze zainwestowany.

O zdarzeniach możesz myśleć w następujący sposób: JEŻELI TO, WTEDY ZRÓB TAMTO.

Pierwsza część wyrażenia - JEŻELI TO, oznacza przyczynę, czyli coś wydarzyło się w systemie, na co ma on zareagować w określony sposób. W tej części definicji zdarzenia można określić także dodatkowe warunki, aby zdarzenie przeszło do fazy wykonania akcji.

Druga część wyrażenia - WTEDY ZRÓB TAMTO, to definicja akcji, jakie mają być wykonane przez system DOMIQ w reakcji na wystąpienie zdarzenia.

Przestuduj poniższe przykłady, które w sposób opisowy przedstawiają koncepcję:



1. *Jeżeli została naruszona czujka ruchu w korytarzu -> załącz oświetlenie w korytarzu.*
2. Powyższy przykład, ale rozszerzony o dodatkowy warunek: *Jeżeli została naruszona czujka ruchu w korytarzu i jest po zmroku -> załącz oświetlenie w korytarzu.*
3. Jeszcze jedna wersja pierwszego przykładu: *Jeżeli została naruszona czujka ruchu w korytarzu i jest uzbrojony alarm -> Włącz wszystkie światła, wyślij notyfikację na urządzenia mobilne, włącz nagrywanie w kamerach.*

Zakładka **Zdarzenia** podzielona jest na trzy sekcje:

1. Pasek przycisków;
2. Struktura;
3. Szczegóły;
4. Akcje;

### Pasek przycisków

W górnej części okna znajduje się pasek z przyciskami.

- **+ Zdarzenie** - Kliknij, aby dodać nowe zdarzenie.
- **+ Grupa** - Kliknij, aby dodać nową grupę zdarzeń.
- **Kopiuj** - Kopiuje do schowka aktualnie zaznaczony element (zdarzenie lub grupę).
- **Wklej** - Wkleja zawartość schowka.
- **Usuń** - Kliknij, aby usunąć zaznaczone elementy.
- **Importuj** - Umożliwia zaimportowanie pliku .xml z konfiguracją.
- **Eksportuj** - Umożliwia zapis do pliku .xml bieżącej listy zdarzeń.
- **Eksportuj zaznaczone** - Umożliwia zapis do pliku jedynie elementów, które mają zaznaczone pole wyboru. Częstą sytuacją jest powielanie danej funkcjonalności w kolejnych instalacjach. W ten sposób możesz wyeksportować tylko część zdarzeń i użyć ich w innej lokalizacji.
- **Zapisz** - Zapisuje bieżącą konfigurację w pamięci modułu Base.
- **Przywróć** - Wczytuje do drzewa struktury ostatnio zapisaną konfigurację.

### Struktura

W oknie **Struktury** wyświetlana jest lista zdarzeń. Lista ma budowę hierarchiczną, tak jak w przypadku zakładki **Remote**.

- Aby dodać nowe elementy użyj przycisków **+ Zdarzenie** lub **+ Grupa**. Jeżeli chcesz dodać element wewnątrz grupy, wybierz ją z listy, a następnie użyj przycisków **+ Zdarzenie** lub **+ Grupa**.

## 8. Automatyzacja

- Aby zmienić pozycję elementu w strukturze, przeciągnij wybrany element i upuść w nowym miejscu. W ten sposób można również dodawać pojedyncze zdarzenia do grup.
- Jeżeli element nie może być dodany w wybranym miejscu, wówczas zobaczysz czerwony znacznik. Dla prawidłowych operacji znacznik ma kolor zielony.
- W drzewie struktury obowiązują następujące zasady dotyczących hierarchii elementów:
  - Na głównym poziomie można elementy obu rodzajów.
  - Wewnątrz **Grupy** można dodać oba rodzaje elementów.
  - Zdarzenia nie mogą mieć elementów zagnieżdżonych.
- Aby zmienić właściwości wybranego elementu, należy kliknąć na nim - jego atrybuty zostaną wyświetlone w oknie **Szczegóły** i **Akcje** (tylko zdarzenia).
- Aby usunąć element(y), należy zaznaczyć go przez kliknięcie w pole wyboru po prawej stronie etykiety, a następnie kliknąć przycisk **Usuń** lub klawisz **backspace** na klawiaturze. Jeśli usuwany element interfejsu ma zagnieżdżone elementy, wówczas one również zostaną usunięte.
- Aby skopiować element, kliknij na nim (po prawej stronie etykiety pojawi się znacznik błękitny prostokąt), a następnie kliknij przycisk **Kopiuj** lub kombinację klawiszy **ctrl + c**. Element został skopiowany i znajduje się w schowku.
- Aby wkleić skopiowany element, użyj przycisku **Wklej** lub skorzystaj ze skrótu klawiaturowego **ctrl + v**. Skopiowany element zostanie umieszczony w następującej lokalizacji:
  - poniżej aktualnie wybranego elementu;
  - na końcu listy, jeżeli zaznaczony jest element typu **Grupa**.
- Pole wyszukiwania pozwala na szybkie wyszukanie elementów w strukturze. Możliwe jest przeszukiwanie na podstawie: opisu, zawartości pola **Kanał** lub typu elementu. Wpisując w pasku wyszukiwania: *E.IDS.input*, zostaną wyświetlone wszystkie kontrolki, które zawierają taką zawartość w polu **Kanał**.
- Pole wyboru **Zaznacz wszystko** umożliwia szybkie zaznaczenie wszystkich elementów struktury. Jest to szczególnie przydatne, kiedy chcesz szybko usunąć całą strukturę.

### 1. Grupa

Grupa służy do łączenia w zbiory zdarzeń wykonujących funkcjonalność o podobnej tematyce. Np. zdarzenia dotyczące systemu alarmowego lub zdarzenia dotyczące wentylacji, itd. W ten sposób łatwiej utrzymać porządek w strukturze zdarzeń.

Grupa może mieć przypisane warunki wykonania. Tworzą one iloczyn logiczny z warunkami poszczególnych zdarzeń. Jednakże warunki przypisane do grupy mają wyższy priorytet - są sprawdzane w pierwszej kolejności. Jeżeli nie są one spełnione, wówczas nie zostanie wyzwolone żadne zdarzenie wewnątrz grupy.

### 2. Zdarzenie

Pojedyncza reguła logiczna.

### Szczegóły

Ta tabela służy do definiowania atrybutów zdarzeń i grup. Na tym etapie wracamy do opisywanej wcześniej koncepcji: JEŻELI TO, WTEDY ZRÓB TAMTO. Okno **Szczegóły** służy do definiowania pierwszej części wyrażenia, czyli JEŻELI TO.

- **Aktywny** - Odznaczone, aby deaktywować zdarzenie. Nieaktywne elementy są wyświetlane w drzewie struktury z przekreśloną etykietą. Domyślnie to pole jest zaznaczone.
- **Opis** - Wprowadź tu zwięzły opis zdarzenia/grupy. Pole opcjonalne.
- **Kanał** (tylko zdarzenia) - **WAŻNE!** Zwróć uwagę na pomarańczowy kolor tła w polu **Kanał** oraz przypis *"Wymaga przedrostka C. lub E."*. Przedrostek **C.** (ang. command) należy rozumieć jako "w reakcji na komendę". Natomiast **E.** (ang. event) jako "w reakcji na zmianę wartości (stanu) identyfikatora". To ważne rozróżnienie, gdyż nie wszystkie komendy pociągają za sobą modyfikację stanu. Zatem w polu **Kanał** należy wpisać nazwę identyfikatora, którego zmiana stanu ma być obserwowana

lub do którego została wysłana komenda.  
Najlepiej zobrazuje to przykład:



1. **Kanał:** *C.IDS.armed.1*

**Dane:** 1

Odebrano komendę uzbrojenia strefy nr 1 w systemie alarmowym. Moduł **Base** może zareagować na to zdarzenie dowolną sekwencją akcji i mogą zostać one wykonane nawet zanim strefa alarmowa zostanie faktycznie uzbrojona.

2. **Kanał:** *E.IDS.armed.1*

**Dane:** 1

Strefa alarmowa nr została uzbrojona - fakt dokonany, stan strefy uległ zmianie. Na tą zmianę stanu moduł **Base** może zareagować dowolną sekwencją akcji np. zgaszenie świateł pozostawionych w domu, obniżenie temperatury ogrzewania itd.

W polu **Kanał** można użyć wzorców dopasowania, dzięki którym stworzysz bardziej zaawansowane definicje zdarzeń. Więcej o tym przeczytasz w akapicie Wzorce w dalszej części rozdziału.

- **Dane** - W tym polu należy wpisać wartość (komendy lub aktualizacji stanu), na którą system ma zareagować. Obrazują to poniższe przykłady:



1. System ma zareagować na włączenie przekaźnika - przyjmuje on wówczas wartość 1

**Dane:** 1

2. System ma zareagować na osiągnięcie temperatury równej 23°C

**Dane:** 23

Jeżeli pole **Dane** pozostanie puste, wówczas zdarzenie zostanie wyzwolone dowolną wartością.

Ponadto w polu **Dane** można wykonać odwołanie do wartości innego identyfikatora używając bindowania identyfikatorów - zostało to przedstawione w akapicie: Bindowanie identyfikatorów.

- **Warunek** - Umożliwia zdefiniowanie dodatkowego warunku, który musi być spełniony, aby zdarzenie zostało wykonane. **Warunek** stanowi iloczyn logiczny z podstawowym warunkiem, jaki zawarty jest w polu **Dane**.  
Podstawową formą warunku jest porównanie bieżącego stanu identyfikatora z określoną wartością, zgodnie z notacją: <nazwa\_identyfikatora><operator><wartość>. Jeżeli wartością identyfikatora jest ciąg znaków (string) np. "*ciemno*", wówczas wartość należy objąć cudzysłowem. Warunków dodatkowych może być kilka, dzięki czemu możliwe jest utworzenie bardziej złożonych reguł logicznych determinujących wykonanie zdarzenia. Należy jednak pamiętać, że komendy poprzedzone warunkiem zostaną wykonane wtedy i tylko wtedy, gdy wyrażenie warunkowe będzie spełnione.

Pole **Warunek** dopuszcza użycie następujących operatorów:

Operator	Opis
<	Mniejszy
>	Większy
<=	Mniejszy równy
>=	Większy równy
==	Równy
~=	Różny
and	Iloczyn logiczny

## 8. Automatyzacja

or	Suma logiczna
not	Negacja

Należy pamiętać, że iloczyn logiczny (and) ma pierwszeństwo przed sumą logiczną (or). W przypadku bardziej złożonych gdzie jednocześnie stosowany jest and i or zalecamy stosować nawiasy, aby wydzielać mniejsze fragmenty instrukcji warunkowej i uniknąć tym samym pomyłki w kolejności sprawdzania wyrażeń (patrz przykłady poniżej).



### Przykłady podstawowe

- `IDS.armed.1==1`  
Tylko wtedy, gdy strefa pierwsza jest uzbrojona.
- `IDS.entry.1==1 and VAR.pora=='lato'`  
Tylko wtedy, gdy jest ustawiony czas na wejście strefy 1 i zmienna `VAR.pora` ma wartość `lato`.
- `MEM.godzina>20 or VAR.pora=='ciemno'`  
Tylko wtedy, gdy zmienna wartość zmiennej `MEM.godzina` jest większa od 20 lub zmienna `VAR.pora` ma wartość `ciemno`.
- `EXP.output.3.1~=1`  
Tylko wtedy, gdy przekaźnik nr 1 w bloku Satel o adresie 3 nie jest włączony.

### Pierwszeństwo w warunkach i wyrażenia złożone

- `zmienna3==1 or zmienna2==0 and zmienna1==1`  
W pierwszej kolejności zostanie sprawdzony iloczyn logiczny, a następnie suma logiczna.
- `(zmienna3==1 or zmienna2==0) and zmienna1==1`  
Dodanie nawiasów do poprzedniego przykładu zupełnie zmienia znaczenie warunku. W pierwszej kolejności sprawdzane jest wyrażenie w nawiasach, a następnie jest realizowany iloczyn logiczny.
- `(zmienna1==1 and zmienna2==2) or (zmienna3==3 and zmienna4==4)`  
Jest to przykład sumy iloczynów logicznych. W pierwszej kolejności sprawdzane są wyrażenia z nawiasów (iloczyn), a następnie rozpatrywana jest suma logiczna. Analogicznie można zrealizować iloczyn sum logicznych itd.

W polu **Warunek** można także użyć funkcji `age`, która zwraca czas wyrażony w sekundach od ostatniej zmiany danego identyfikatora. Dzięki temu zdarzenie może zostać wykonane z określonym opóźnieniem w stosunku do ostatniej zmiany dowolnego identyfikatora w systemie.



1. `age('MODBUS.int.vent.temp') > 60`.  
Taki warunek należy rozumieć jako: "wykonaj, gdy ostatnia zmiana wartości identyfikatora `MODBUS.int.vent.temp` była co najmniej 60 sekund temu".

Ponadto w polu **Warunek** można deklarować wyrażenia używając wzorców (patrz Wzorce) oraz bindowania identyfikatorów (patrz Bindowanie identyfikatorów). W ten sposób możliwe jest zrealizowanie bardzo złożonych reguł, co czyni zdarzenia bardzo ważnym elementem automatyzacji budynku.

## Akcje

W oknie **Akcje** definiowana jest lista czynności, jakie moduł **Base** ma wykonać jako wynik wywołania zdarzenia. Opis interfejsu znajduje się w podrozdziale: 8.1. Akcje



### 1. Wykorzystanie czujników PIR centrali alarmowej do załączania świateł.

Przyjmijmy, że czujnik jest podłączony do wejścia binarnego nr 1 w module Satel INT-IORS z adresem 1, a oświetlenie do przekaźnika nr 3 w tym samym module. Całość sterowana jest przez moduł DOMIQ/Expander.

**Kanał:** *E.EXP.input.1.1*

**Dane:** *1*

**+ Komenda:**

**Nazwa:** *C.EXP.output.1.3*

**Wartość:** *1.*

### 2. Automatyczne podnoszenie rolety w przypadku otwarcia okna.

Przyjmijmy, że czujnik otwarcia okna podłączony jest do centrali alarmowej Satel, nr wejścia 10, a roleta podłączona jest do modułu Shelly 2.5 o nazwie `rolety_salon`

**Kanał:** *E.IDS.input.10*

**Dane:** *1*

**+ Komenda:**

**Nazwa:** *C.SHELLY.rolety\_salon.roller.0*

**Wartość:** *up*

### Zagadnienia zaawansowane

#### Bindowanie identyfikatorów

To funkcjonalność, która upraszcza tworzenie złożonych reguł logicznych. Umożliwia odwołanie się w definicji zdarzenia do bieżącej wartości dowolnego identyfikatora. W tym celu należy nazwę identyfikatora ująć w nawiasy klamrowe, np. {LCN.output.0.10.1}. Gdy użyjesz takiego zapisu, moduł Base automatycznie w momencie wywołania danej reguły, pobierze aktualną wartość wskazanego identyfikatora. Bindowanie identyfikatorów możesz użyć w polach **Dane** w oknie **Szczegóły**, w polu **Wartość** w oknie **+ Komenda**. Ponadto bindowanie identyfikatorów jest dostępne w polu **Warunek**, ale wyjątkowo nie wymaga stosowania nawiasów klamrowych.

Przestuduj poniższe przykłady:



#### 1. Wykonanie komendy w oparciu o wartość wybraną przez użytkownika w aplikacji Remote.

Założmy sytuację, gdzie w reakcji na zdarzenie ma być wykonana pewna komenda. Jednakże wartość komendy ma zależeć od opcji wybranej przez użytkownika w aplikacji Remote. Przykładem może być ustawienie temperatury zadanej ogrzewania (w tym przykładzie regulator LCN), w wyniku uzbrojenia alarmu (co może oznaczać wyjście z domu). Jest to szczególnie przydatne, gdy użytkownik chce mieć zawsze możliwość zmiany tej temperatury w przyszłości bez rekonfiguracji systemu.

Wybór użytkownika jest zapisany np. w zmiennej MEM. Na potrzeby przykładu nazwiemy ją: MEM.userChoice. Wówczas zdarzenie może mieć następującą definicję:

**Kanał:** E.IDS.armed.1

**Dane:** 1

W sekcji **Akcje** -> **+ Komenda**

**Nazwa:** C.LCN.regulator.0.10.1

**Wartość:** {MEM.userChoice}

#### 2. Zapisywanie daty i godziny wystąpienia zdarzenia alarmowego w systemie Satel do zmiennej nieulotnej MEM.

**Kanał:** E.IDS.alarm.1

**Dane:** 1

W sekcji **Akcje** -> **+ Komenda**

**Nazwa:** C.MEM.lastAlarm

**Wartość:** {CLOCK.datetime}

W momencie wystąpienia zdarzenia, moduł Base pobierze aktualną wartość identyfikatora CLOCK.datetime, który przechowuje datę i godzinę i zapisze ją w zmiennej MEM.lastAlarm.

## Arytmetyka

Moduł **Base** umożliwia wykonywanie działań arytmetycznych na wartościach identyfikatorów.

Dostępne są następujące operacje: dodawanie, odejmowanie, mnożenie, dzielenie, potęgowanie (^), dzielenie modulo (%). Arytmetyka dostępna jest w polach **Dane** (zdarzenia) oraz **Warunek** (zdarzenia i timery).



### Zastosowanie działań arytmetycznych w polu **Warunek** do sterowania ogrzewaniem z histerezą.

Na potrzeby przykładu założmy, że: temperatura zadana jest przechowywana w zmiennej MEM o nazwie `MEM.temp.salon`. Natomiast temperatura mierzona jest dostępna w identyfikatorze `MODBUS.int.sensor.temp`. Wyjście temperatury mierzonej poza ustalony zakres ma przełączyć przełącznik sterujący zaworem pętli grzewczej. Założmy, że odchyłka temperatury mierzonej może wynosić maksymalnie 1,5°C. Przełącznik sterujący zaworem jest podłączony do bloku Satel INT-IORS, sterowanego modułem DOMIQ/Expander. Blok ma adres 1, a przełącznik nr 3.

- **Temperatura za niska**

**Kanał:** `E.MODBUS.int.sensor.temp`

**Dane:** pozostaw puste, aby wywołać zdarzenie przy każdorazowej zmianie temperatury pomieszczenia.

**Warunek:** `MODBUS.int.sensor.temp - 1,5 <= MEM.temp.salon`

**Ważne!** Jest to jednocześnie przykład bindowania identyfikatorów i działań arytmetycznych w polu **Warunek**.

W sekcji **Akcje** -> **+ Komenda**

**Nazwa:** `C.EXP.output.1.3`

**Wartość:** `1`

Analogicznie można zrealizować działanie dla temperatury za wysokiej, zmieniając jedynie operator w warunku na `>=` oraz wartość komendy na `0`.

Użycie działań arytmetycznych w polu **Dane** jest ściśle związane z bindowaniem identyfikatorów. Dzięki temu możesz tworzyć bardziej elastyczne definicje zdarzeń, które będą wywoływane w zależności od stanu innego identyfikatora w module Base. **Pamiętaj, aby nazwę identyfikatora w tym przypadku objąć nawiasami klamrowymi.**

Także działania arytmetyczne używane w polu **Wartość** w oknie **+ Komenda** wymaga użycia nawiasów klamrowych. Arytmetyka w tym miejscu pozwala na dynamiczną modyfikację wartości komendy w zależności od stanu innego identyfikatora. Dzięki temu jest możliwe jest np. skalowanie wartości.



### 1. Kompensacja temperatury mierzonej.

Niektóre czujniki temperatury wymagają kompensacji swoich odczytów ze względu na miejsce ich montażu. Założmy, że chcemy skompensować odczyty czujnika podłączonego do magistrali MODBUS o 2°C i zapisać nową skompensowaną wartość w zmiennej VAR o nazwie `VAR.sensor.temp`. Przyjmijmy nast. identyfikator czujnika: `MODBUS.int.sensor.temp`.

**Kanał:** `E.MODBUS.int.sensor.temp`

**Dane:** Pozostaw puste, aby zdarzenie było wyzwalane przy każdej zmianie temperatury.

**+Komenda:**

**Nazwa:** `C.VAR.sensor.temp`

**Wartość:** `$D0 - 2`

Powyższy zapis należy rozmieścić jako: weź wartość z jaką zostało wywołane zdarzenie i pomniejsz o 2.

Więcej o użyciu parametrów przeczytasz w sekcji Parametry.



## 8. Automatyzacja

### Wzorce

Wzorce umożliwiają tworzenie bardzo uniwersalnych reguł, które mogą reagować na wiele zdarzeń jednocześnie. Jest to bardzo duże uproszczenie konfiguracji, znaczna oszczędność czasu, a także optymalizacja z punktu widzenia wydajności pracy systemu. Wzorce można stosować wyłącznie w polu **Kanał**. Można użyć kilku wzorców w obrębie jednej definicji zdarzenia.

Wzorzec	Opis
%d	Dowolna cyfra od 0 do 9.
%a	Dowolna litera alfabetu.
%w	Dowolna litera lub cyfra. Parametr %w jest połączeniem parametrów %d oraz %a .
%.	Dowolny znak.
+	Jedno lub więcej wystąpień. Parametr ten jest dodatkiem do pierwszych czterech parametrów.
*	Zero lub więcej wystąpień. Parametr ten jest dodatkiem do pierwszych czterech parametrów.
[0-9aZ]	Dowolna cyfra lub litera z określonego zakresu. Zakresy można swobodnie zmieniać, np. [1-4aD]. Ten parametr pozwala na wygodne filtrowanie zdarzeń. Umożliwia filtrowanie występowania jednocześnie cyfr i liter.

Aby zrozumieć koncepcję wzorców rozpatrz poniższe przykłady:



1. Załóżmy sytuację, że system DOMIQ ma zareagować taką samą akcją, na naruszenie jednej z  $n$  czujek ruchu, gdzie  $n$ , to dowolna liczba czujek. Zamiast tworzyć zdarzenie dla każdej czujki, możesz utworzyć jedno zdarzenie korzystając z wzorców, które wychwyci wszystkie te zdarzenia! Załóżmy, że czujki są podłączone do modułu Satel INT-IORS o adresie 3, sterowanym przez moduł DOMIQ/Expander. Wówczas zdarzenie miałoby nast. definicję:  
**Kanał:** *E.EXP.input.3.(%d)*  
Zdarzenie zostanie wywołane dla dowolnego wejścia binarnego (1-8), które zmieniło swoją wartość z 0 na 1.  
**Dane:** 1  
W sekcji **Akcje** zadeklaruj komendy.
2. Zdarzenie zostanie wywołane, jeżeli którykolwiek przekaźnik w module LCN o adresie 10 zostanie wyłączony.  
**Kanał:** *E.LCN.relay.0.10.(%d)*  
**Dane:** 0
3. Zdarzenie zostanie wywołane, jeżeli którekolwiek wyjście ściemniane w segmencie nr 10 zostanieysterowane na 100%.  
**Kanał:** *E.LCN.output.10.(%d+).( %d)*  
**Dane:** 100
4. Zdarzenie zostanie wykonane, jeżeli jakiegokolwiek urządzenie alarmowe zmieni swój stan - w reakcji zarówno na zmiany stanu identyfikatorów typu: *IDS . alarm . x*, jak i *IDS . output . x . itd.*  
**Kanał:** *E.IDS.(%a+).( %d+)*  
**Dane:** 1



## Parametry

Parametry są ściśle związane z wzorcami, ponieważ przechwytyją elementy wyodrębnione przez wzorce. Można użyć kilka parametrów w obrębie definicji zdarzenia. Aby zawartości wzorców były przekazywane do parametrów, wzorec w definicji zdarzenia musi być objęty nawiasami okrągłymi. Dzięki parametrom utworzysz bardzo uniwersalne zdarzenia (patrz przykłady).

Parametry mogą być użyte w polu **Warunek** oraz w polach **Nazwa** i **Wartość** w oknie edycji **+ Komenda**.

Parametr	Opis
\$C0	Przechowuje pełną nazwę zdarzenia (wraz z identyfikatorem).
\$C1-9	Przechowuje zawartość wyodrębnioną przez wzorec. Numeracja parametrów jest nadawana w zależności od użytej liczby wzorców. Jeżeli wykorzystane są 2 wzorce, wówczas dostępne są parametry \$C1 oraz \$C2 itd.
\$D0	Przechowuje wartość z jaką zdarzenie zostało wywołane.
\$D1-9	Przechowuje wartość elementu wyodrębnionego przez wzorec. Ten rodzaj parametrów może być stosowany jedynie w stosunku do wzorców użytych w polu <b>Dane</b> . Numeracja parametrów jest nadawana w zależności od użytej liczby wzorców. Jeżeli wykorzystane są 2 wzorce, wówczas dostępne są parametry \$D1 oraz \$D2 itd.

**Parametr \$C0 zawsze przechowuje pełną nazwę zdarzenia, natomiast \$D0 jego aktualną wartość.**

Parametr \$D0 daje duże możliwości, dlatego ważne jest zrozumienie sposobu jego użycia.



### 1. Zdarzenie reagujące na przekroczenie obserwowanej wartości np. przekroczenie temperatury w pomieszczeniu.

Na potrzeby przykładu założmy, że obserwujemy temperaturę mierzoną przez czujnik podłączony do magistrali MODBUS. Jego identyfikator to `MODBUS.int.sensor.temp`. Zdarzenie ma wykonać określoną listę akcji, kiedy temperatura spadnie poniżej 0°C.

**Kanał:** `E.MODBUS.int.sensor.temp`

**Dane:** Pozostaw to pole puste, aby zdarzenie było wywołane przez każdą zmianę mierzonej temperatury.

**Warunek:** `$D0 < 0`

Warunek należy rozumieć następująco: *jeżeli wartość, z jaką zostało wywołane zdarzenie jest mniejsza od 0.*

### 2. Załączanie przekaźnika w zależności od stanu wejścia binarnego.

Bez użycia wzorców i parametrów wymagałoby to utworzenia dwóch zdarzeń: jednego dla logicznej 1 na wejściu binarnym i drugiego dla logicznego 0. Z ich użyciem, można zrealizować to przy pomocy pojedynczego zdarzenia. Założmy, że obserwujemy zmianę stanu wejścia binarnego nr 5 podłączonego do modułu Satel INT-IORS o adresie 2, sterowanego przez moduł Expander oraz, że przekaźnik ma nr 7 i jest w module LCN o adresie 10.

**Kanał:** `E.EXP.input.2.5`

**Dane:** Pozostaw to pole puste, aby zdarzenie było wyzwalane przez oba stany wejścia binarnego.

**+ Komenda:**

**Nazwa:** `C.LCN.relay.0.10.7`

**Wartość:** `$D0`

Użycie parametru `$D0` przechwytyje wartość wejścia binarnego (1 lub 0) i przekazuje tą wartość do przekaźnika.

### 3. Zapisanie do zmiennej MEM numeru czujki alarmowej, która wywołała alarm.

Zmienna MEM będzie zachowywać informację o ostatnio naruszonej czujce alarmowej. Na potrzeby przykładu nazwiemy ją: `MEM.lastInput`. To przykład, w którym pokazujemy użycie parametrów w polu **Kanał**.

**Kanał:** `E.IDS.input. (%d+)`

**Dane:** `1`

**+ Komenda:**

**Nazwa:** `C.MEM.lastInput`

**Wartość:** `$C1`

Wzorzec `(%d+)` oznacza "jedna lub więcej cyfr". Zatem przechwycona zostanie zarówno cyfra od 0-9, jak i liczba np. 123.

Wartość przechwycona przez wzorzec dostępna jest następnie w parametrze `$C1`. Zatem odwołując się do niego w komendzie powodujemy zapisanie numeru czujki, jako wartości zmiennej MEM.

### 4. Wysłanie notyfikacji do aplikacji DOMIQ/Remote.

**Kanał:** `E.IDS.alarm. (%d+)` - sprawdzana będzie dowolna strefa alarmowa.

**Dane:** `1`

**+ Komenda:**

**Wartość:** `C.REMOTE.notify`

**Wartość:** Tu wpisz treść powiadomienia, która zostanie wyświetlona na ekranie urządzenia. Wpisując np. *Alarm w strefie \$C1*, użytkownik w przypadku alarmu zobaczy komunikat o treści *Alarm w strefie x*, gdzie x to nr strefy, w której wystąpił alarm.

### 5. Kanał: `E.LCN.output.0.10. (%d)`

**Dane:** `(%d+)`

Założmy, że wyjście ściemniane nr 2 w module LCN o adresie 10 osiąga wartość 50%. Generowane jest wtedy zdarzenie `E.LCN.output.0.10.2=50`. Wówczas do parametru `$C1` przypisywany jest numer wyjścia, które zmieniło stan, w tym przypadku 2, natomiast parametr `$D1` ma wartość 50.

## 8.3. Timery

Zakładka **Timery** służy do definiowania reguł czasowych (dalej zwane timerami), które mają być wykonane przez moduł Base w określonym czasie. Dla każdej reguły można określić dokładną datę i czas wykonania, dodatkowe warunki oraz listę akcji, które będą wykonane przy każdorazowym wywołaniu timera. Dzięki timerom możliwe jest zautomatyzowanie powtarzalnych czynności, dzięki czemu pamiętanie o nich przestaje być konieczne.

Zakładka **Timery** w swojej budowie jest identyczna, jak zakładka **Zdarzenia**. Znajdziesz w niej listę reguł - **Struktura**, formularz z parametrami timera - **Szczegóły** oraz interfejs do definiowania akcji - **Akcje**.

Omówimy okno **Szczegóły**, gdyż ono różni się od tego z zakładki **Zdarzenia**, opis pozostałych elementów jest tożsamy z tym przedstawionym dla zakładki **Zdarzenia**.

### Szczegóły

Sekcja **Szczegóły** służy do definiowania atrybutów timerów i grup.

Grupy mają trzy atrybuty:

- **Aktywny** - Odznacz, aby deaktywować grupę. Nieaktywne elementy są wyświetlane w drzewie struktury z przekreśloną etykietą. Domyślnie to pole jest zaznaczone.
- **Opis** - Krótki opis grupy, który jest wyświetlany w **Strukturze**.
- **Warunek**: Warunek dodany do grupy timerów tworzy iloczyn logiczny z warunkami poszczególnych timerów. Zatem jeżeli warunek grupy nie zostanie spełniony, wówczas nie zostanie wyzwolony żaden timer przypisany do grupy.

Każdy timer ma natomiast zestaw dziewięciu atrybutów.

- **Aktywny** - Odznacz, aby deaktywować timer. Nieaktywne elementy są wyświetlane w drzewie struktury z przekreśloną etykietą. Domyślnie to pole jest zaznaczone.
- **Opis** - Krótki opis timera, który jest wyświetlany w **Strukturze**.
- **Miesiąc** - Miesiąc w przedziale od **1** do **12**. Jeżeli pole zostanie puste, a pole **Dzień** zostanie uzupełnione, wówczas timer będzie wywoływany w każdym miesiącu w roku, w wybrany dzień.
- **Dzień** - Dzień miesiąca w przedziale od **1** do **31**. Jeżeli pole zostanie puste, wówczas timer będzie wywoływany codziennie.
- **Godzina** - Godzina w przedziale od **0** do **23**. **Nie należy wpisywać tu wartości poprzedzonych zerem np. 06, 02 itp.**
- **Minuta** - Minuta w przedziale od **0** do **59**. Jeżeli pole **Minuta** jest uzupełnione, natomiast pole **Godzina** pozostanie puste, wówczas timer będzie wyzwalany co godzinę, w wybranej minucie. Natomiast jeżeli pole **Minuta** pozostaje puste, a pole **Godzina** jest uzupełnione, wówczas timer będzie wykonywany co minutę przez całą wskazaną godzinę.

**Nie należy wpisywać tu wartości poprzedzonych zerem np. 06, 02 itp.**

- **Sekunda** - Sekunda w przedziale od **0** do **59**. Jeżeli pole **Sekunda** jest uzupełnione, natomiast pola **Godzina** i **Minuta** pozostaną puste, wówczas timer będzie wyzwalany co minutę w wybranej sekundzie. Uwaga! Jeżeli wprowadzanych jest kilka pozycji, wartości muszą się różnić o co najmniej 2, np. 34, 36, 38 itd. Jeżeli chcesz, aby zdarzenie było wywoływane co minutę o pełnej minucie, wówczas w polu **Sekunda** wpisz: **0**. **Nie należy wpisywać tu wartości poprzedzonych zerem np. 06, 02 itp.**
- **Dzień tygodnia** - Dzień tygodnia w przedziale od **1** do **7**, gdzie **1** odpowiada niedzieli, **2** poniedziałkowi, itd. Pozostaw to pole puste, jeżeli chcesz, aby timer był wywoływany każdego dnia tygodnia. Z użyciem tego pola łatwo jest utworzyć np. osobne reguły dla dni powszednich i weekendów.
- **Warunek** - Dodatkowy warunek, który musi być spełniony, aby timer został wzbudzony. Obowiązują tutaj identyczne zasady, jak w przypadku **Zdarzeń**.

W każdym z powyższych pól można podać więcej niż jedną wartość, używając przecinka jako separatora (poza polem **Warunek**) pomiędzy kolejnymi wartościami. Jeżeli pola są ze sobą powiązane, wówczas utworzone zostają kombinacje z zawartości pól. Przystuduj poniższe przykłady:

## 8. Automatyzacja



- W polu **Godzina** wpisano *16,20*, natomiast w polu **Minuta** *15,45*. W takim przypadku timer będzie wywoływany o 16:15, 16:45, 20:15 oraz 20:45.
- W polu **Dzień** wpisano *1,2,3,4,5,6,7*, natomiast w polu **Dzień tygodnia**: *2,3,4,5,6*. W tym przypadku timer będzie wywołany w dni powszednie, w pierwszym tygodniu miesiąca.
- W polu **Dzień** wpisano *1*, w polu **Godzina**: *6*, a w polu **Minuta**: *0*. Timer będzie wówczas wykonany każdego pierwszego dnia miesiąca o godzinie 6:00.

### Zegar astronomiczny

Moduł Base umożliwia wykonywanie timerów w oparciu o godziny wschodu i zachodu słońca (zegar astronomiczny). W ten sposób utworzysz reguły np. załączające oświetlenie po zmroku nie angażując do tego żadnych dodatkowych czujników.

Aby zegar astronomiczny funkcjonował niezbędne jest uzupełnienie danych w sekcji **Lokalizacja** w zakładce **Ustawienia**.

Aby zdefiniować timer z użyciem zegara astronomicznego użyj jednego ze słów kluczowych:

- *sunset* - Wprowadź tą wartość w polu **Godzina**, aby timer był wykonany dokładnie w godzinie i minucie zachodu słońca.
- *sunrise* - Wprowadź tą wartość w polu **Godzina**, aby timer był wykonany dokładnie w godzinie i minucie wschodu słońca.

Możliwe jest również wykonywanie timerów przesuniętych w czasie względem godziny wschodu i zachodu słońca. W tym celu należy wprowadzić przesunięcie w polu **Minuta**. Przesunięcie może mieć wartość dodatnią lub ujemną. Możliwe jest użycie wielu wartości przesunięć w tej samej definicji timera. Wówczas ich wartości należy wpisać w polu **Minuta** rozdzielone przecinkiem.



- W polu **Godzina** wpisano: *sunrise*, a w polu **Minuta**: *30* - timer zostanie wykonany 30 minut po wschodzie słońca.
- Analogicznie, jeżeli w polu **Godzina** wpisano: *sunrise*, a w polu **Minuta**: *-30*, timer zostanie wykonany 30 minut przed wschodem słońca.
- Identyczne reguły mają zastosowanie dla wyrażenia *sunset*.
- Jeżeli w polu **Minuta** zostało wpisane kilka wartości np. *15,30,45*, wówczas timer zostanie wywołany trzykrotnie dla każdego przesunięcia osobno.

### Timery wywoływane komendami

Obok timerów uruchamianych w oparciu o definicje w zakładce **Timery**, istnieje również możliwość utworzenia timerów przy użyciu komend. Jest to bardziej dynamiczna forma tworzenia timerów, która może być efektem np. naciśnięcia przycisku, wywołania zdarzenia lub akcją wywołaną przez inny timer.

W module Base są dwa rodzaje komend do tworzenia timerów dynamicznie: timer z referencją i timer bez referencji.

#### 1. TIMER (timer z referencją)

Komenda `TIMER` służy do dynamicznego tworzenia timera, który wykona określoną czynność po wskazanym czasie. Ten rodzaj komendy wymaga nadania timerowi nazwy unikalnej w całym systemie - dlatego nazywany jest timerem z referencją, gdyż nazwa stanowi referencję do timera, umożliwiającą jego anulowanie.

Każdorazowe wywołanie komendy `TIMER` z tą samą nazwą, usuwa poprzednią instancję timera i tworzy nową, rozpoczynając odliczanie czasu od początku. Możliwe jest również anulowanie timera poprzez ponowne wywołanie go z czasem równym *0*.

Przykładowym zastosowaniem dla komendy `TIMER` jest implementacja oświetlenia schodowego, automatyczne uzbrojenie alarmu itp.

Składnia komendy jest następująca:

`TIMER.nazwa.opóźnienie.komenda=dane`, gdzie

`nazwa` - Nazwa timera (bez polskich znaków i spacji), która jest unikalna wśród innych komend typu. `TIMER`

`opóźnienie` - Opóźnienie po którym zostanie wykonana akcja przypisana do tego timera. Opóźnienie może być wyrażane w godzinach, minutach oraz sekundach oraz milisekundach, kolejno skróty: h, m, s, ms.

Opóźnienie może być również dynamicznie zczytywane z bieżącej wartości dowolnego identyfikatora w systemie DOMIQ. Dzięki temu możliwe jest użycie jednej definicji do wywołania timerów z różnymi opóźnieniami. Zmiana długości opóźnienia bazuje na wartości wskazanej zmiennej. Składnia poleceń: `TIMER.<nazwa>.{nazwa_identyfikatora}<jednostka>.<komenda>=<wartość>;`

`komenda` - Akcja, która zostanie wykonana po upływie czasu odliczania.

`wartość` - Wartość jaką ma przyjąć identyfikator, do którego wysyłana jest komenda. W przypadku zdarzeń jest wartością, z jaką zdarzenie zostanie rozgłoszone.

## 8. Automatyzacja

Przestuduj następujące przykłady:



### 1. Wyłączenie oświetlenia po określonym czasie.

Klasycznym przykładem dla takiego scenariusza jest sterowanie oświetleniem załączanym z czujki ruchu, które ma zostać automatycznie wyłączone w przypadku braku ruchu. Włączenie oświetlenia i wywołanie komendy `TIMER` jest efektem wykonania zdarzenia, dlatego ono również zostanie zaprezentowane w tym przykładzie.

Żałujemy, że czujka ruchu podłączona jest do wejścia nr 1 modułu Satel INT-IORS o adresie 1, sterowanego przez moduł DOMIQ/Expander. Natomiast oświetlenie jest sterowane przekaźnikami nr 3 w tym samym module.

**Kanał:** `E.EXP.input.1.1`

**Dane:** `1`

Następnie należy dodać dwie komendy z użyciem opcji **+ Komenda:**

Komenda 1 - załączenie oświetlenia

**Nazwa:** `C.EXP.output.1.3`

**Wartość:** `1`

Komenda 2 - uruchomienie timera na zgaszenie światła po 2 minutach.

**Nazwa:** `TIMER.czujka_korytarz.2m.C.EXP.output.1.3`

**Wartość:** `0`

Komendę nr 2 należy rozumieć następująco: *utwórz timer o nazwie czujka\_korytarz, który po 2 minutach wywoła komendę C.EXP.output.1.3 z wartością 0.*

### 2. Automatyczne uzbrojenie alarmu.

Jest to przykład zdarzenia, które w przypadku niewykrycia ruchu przez żadną z czujek przez określony czas, uzbroi strefy w centrali Satel. Takie zdarzenie może służyć jako zabezpieczenie na wypadek zapomnienia o uzbrojeniu alarmu podczas wychodzenia z domu.

**Kanał:** `E.IDS.input.(%d+)`

**Dane:** `1`

**+ Komenda:**

**Nazwa:** `TIMER.autoalarm.2h.C.ALARMON`

**Wartość:** `1`

Zdarzenie wychwyci każde naruszenie czujki ruchu i w efekcie utworzy timer o nazwie `autoalarm`, który po upływie 2 godzin wykona komendę `C.ALARMON` z wartością `1`.

Dla komendy `C.ALARMON` należy utworzyć osobne zdarzenie, które w rezultacie wykona komendy uzbrojenia poszczególnych stref alarmowych.

**Kanał:** `C.ALARMON`

**Dane:** `1`

Następnie należy komendy z użyciem opcji **+ Komenda** w liczbie odpowiedniej do liczby stref, które mają zostać uzbrojone. Poniżej przykładowa pojedyncza komenda:

**Nazwa:** `C.IDS.armed.1`

**Dane:** `1;pin:12345`



### 3. Automatyczne wyłączenie oświetlenia w pokoju dziecięcym z użyciem aplikacji Remote.

Założmy, że światło w pokoju dziecięcym ma być włączone (30% jasności) przez 30 minut, po czym ma automatycznie zgasnąć. Światło jest sterowane przez wyjście ściemniane nr 2 w module LCN o adresie 50. Cała procedura ma być inicjowana w aplikacji Remote. W tym celu zastosowana zostanie kontrolka **Przycisk**. Jej definicja jest następująca:

W zakładce **Krótko** należy dodać dwie komendy.

#### Komenda 1:

Kliknij **+ LCN**

**Moduł:** 50

**Wyjście:** 2

**Wartość:** 30

#### Komenda 2:

Kliknij **+ Komenda:**

**Nazwa:** *TIMER.pokoj\_dzieciecy.30m.C.LCN.output.0.50.2*

**Wartość:** 0

### 4. Anulowanie timera.

Ten przykład prezentuje anulowanie timera z użyciem komendy. Komenda może być efektem wykonania zdarzenia, innego timera, naciśnięcia przycisku w aplikacji Remote, kontrolki na wizualizacji itd. Na potrzeby przykładu założmy, że jest to efekt wykonania zdarzenia. Wówczas należy dodać następującą komendę (**+ Komenda**):

**Nazwa:** *TIMER.alarm1.0*

**Wartość:** 0

W wyniku tej komendy zostanie anulowany timer o nazwie `alarm1`.

### 5. Timer z dynamicznie ustalaniem opóźnieniem

Na potrzeby przykładu założmy, że jest to efekt wykonania zdarzenia. Wówczas należy dodać następującą komendę (**+ Komenda**):

**Nazwa:** *TIMER.test.{VAR.test}s.C.LCN.output.0.10.1*

**Wartość:** 100

Ten przykład należy czytać następująco - wywołaj timer o nazwie *test* z opóźnieniem równym wartości zmiennej `VAR.test`. Po upływie wartości opóźnienia włącz na 100% wyjście nr 1 w module LCN o adresie 10. Analogicznie można używać nazw identyfikatorów w komendach `DELAY`.

## 2. DELAY (timer bez referencji)

Komenda `DELAY` również służy do dynamicznego tworzenia timera, który wykona określoną czynność po wskazanym czasie. Jednakże w odróżnieniu od komendy `TIMER` nie umożliwia anulowania bieżącego odliczania. Każdorazowe wywołanie komendy `DELAY` tworzy nową instancję timera, który po upływie określonego czasu wykona przypisaną czynność.

`DELAY.opóźnienie.komenda=dane`, gdzie

`opóźnienie` - Opóźnienie po którym zostanie wykonana akcja przypisana do tego timera. Opóźnienie może być wyrażane w godzinach, minutach oraz sekundach oraz milisekundach, Kolejno skróty: h, m, s, ms.

Opóźnienie może być również dynamicznie zczytywane z bieżącej wartości dowolnego identyfikatora w systemie DOMIQ. Dzięki temu możliwe jest użycie jednej definicji do wywołania timerów z różnymi opóźnieniami. Zmiana długości opóźnienia bazuje na wartości wskazanej zmiennej. Składnia poleceń: `DELAY.{nazwa_identyfikatora}<jednostka>.<komenda>=<wartość>;`

`komenda` - Akcja, która zostanie wykonana po upływie odliczania.

`wartość` - Wartość jaką ma przyjąć identyfikator, do którego wysyłana jest komenda. W przypadku zdarzeń jest `wartość`, z jaką zdarzenie zostanie rozgłoszone.



### Zagadnienia zaawansowane

#### Bindowanie identyfikatorów

Bindowanie identyfikatorów jest dostępne także dla timerów. Umożliwia odwołanie się w definicji timera do bieżącej wartości dowolnego identyfikatora. Gdy użyjesz takiego zapisu, moduł Base automatycznie w momencie wywołania danej reguły, pobierze aktualną wartość wskazanego identyfikatora.

Bindowanie identyfikatorów możesz użyć w oknie + **Komenda** w sposób identyczny, jak zostało to przedstawione dla zdarzeń.

Użycie bindowania identyfikatorów w polach formularza **Szczegóły** umożliwia tworzenie bardziej interaktywnych timerów. Przykładem tego mogą być reguły, których czas uruchomienia jest zależny od opcji wybranych przez użytkownika w aplikacji Remote. Bindowanie identyfikatorów można użyć we wszystkich polach formularza **Szczegóły** poza polem **Opis** i **nie jest wymagane tu zastosowanie nawiasów klamrowych**.



#### 1. Podnoszenie rolet o godzinie wybranej przez użytkownika

Założmy scenariusz, w którym rolety w budynku mają się podnosić o określonej godzinie, którą użytkownik systemu ustala samodzielnie w aplikacji Remote.

W tym celu należy utworzyć kontrolkę **Czas** w zakładce **Remote**. Nastawy powinny być odporne na restarty modułu, dlatego w polach **Kanał dla godzin** i **Kanał dla minut** należy wpisać nazwy identyfikatorów, będących zmiennymi typu MEM. W naszym przykładzie: *MEM.roletyH* oraz *MEM.roletyM*.

Następnie możemy użyć tych identyfikatorów w definicji timera:

**Godzina:** *MEM.roletyH*

**Minuta:** *MEM.roletyM*

W sekcji **Akcje** należy zdefiniować komendy sterujące roletami.

#### 2. Uruchomienie timera w dni tygodnia wybrane przez użytkownika

Jest to rozwinięcie przykładu nr 1. Dodatkowo założmy, że użytkownik chce mieć kontrolę na tym, w które dni tygodnia reguła ma być wykonywana.

W tym celu należy zdefiniować kontrolkę **Wybór** w zakładce **Remote** wg przykładu przedstawionego w opisie tej kontrolki. W polu **Kanał** należy wpisać nazwę identyfikatora będącego zmienną MEM, gdyż chcemy również, aby nastawy były zapamiętywane w pamięci nieulotnej. W naszym przykładzie: *MEM.roletyDni*

W definicji timera przedstawionej w przykładzie 1, należy dodatkowo uzupełnić pole **Dzień tygodnia** wpisując *MEM.roletyDni*.

#### 3. Warunkowe wykonanie timera na podlewanie ogrodu.

Założmy, że timer ma cyklicznie uruchamiać system podlewania ogrodu o 4:00 oraz o 20:00, każdego dnia tygodnia. Jednakże ma być on również uzależniony od czujnika deszczu oraz mieć możliwość ręcznej deaktywacji przez użytkownika z poziomu aplikacji Remote. Założmy, że czujnik deszczu jest dostępny jako identyfikator: *MODBUS.int.sensor.rain*.

*MODBUS.int.sensor.rain*.

##### Ręczna deaktywacja

W zakładce **Remote** dodaj **Przełącznik**. Aby ustawienia były odporne na restarty urządzenia, wpisz identyfikator wskazujący zmienną MEM, np. *MEM.autopodlewanie*. Przełącznik w zależności od stanu nada zmiennej wartość: *on* lub *off*.

Następnie zdefiniuj timer:

**Godzina:** 4, 20

**Minuta:** 0

**Dzień tygodnia:** Pozostaw to pole puste, aby timer wykonywał się każdego dnia.

**Warunek:** *MEM.autopodlewanie=='on' and MODBUS.int.sensor.rain == 0*

W sekcji **Akcje** należy zdefiniować komendy inicjujące podlewanie.



## Arytmetyka

Moduł Base umożliwia wykonywanie działań arytmetycznych na wartościach identyfikatorów.

Dostępne są następujące operacje: dodawanie, odejmowanie, mnożenie, dzielenie, potęgowanie (^), dzielenie modulo (%). Arytmetyka dostępna jest w polach: **Warunek** oraz **Wartość** w oknie **+ Komenda**.



### **Automatyczne uruchomienie ogrzewania pod warunkiem, że temperatura mierzona jest niższa od wartości zadanej o określoną wartość.**

Na potrzeby przykładu założmy, że: temperatura zadana jest przechowywana w zmiennej MEM o nazwie `MEM.temp.salon`. Natomiast temperatura mierzona jest dostępna w identyfikatorze `MODBUS.int.sensor.temp`. Ogrzewanie ma być załączone o określonej porze, ale wyłącznie gdy różnica temperatury pomieszczenia i zadanej jest na określonym poziomie (1,5°C lub większa).

Jeżeli warunek jest spełniony, wówczas załączony zostanie przełącznik sterujący zaworem pętli grzewczej. Przełącznik jest podłączony do bloku Satel INT-IORS, sterowanego modułem DOMIQ/Expander. Blok ma adres 1, a przełącznik nr 3.

**Godzina:** 5

**Minuta:** 0

**Warunek:** `MODBUS.int.sensor.temp - 1,5 <= MEM.temp.salon`

**Ważne!** Jest to jednocześnie przykład bindowania identyfikatorów i działań arytmetycznych w polu **Warunek**.

W sekcji **Akcje** -> **+ Komenda**

**Nazwa:** `C.EXP.output.1.3`

**Wartość:** 1

Użycie arytmetyki w oknie **+ Komenda** zostało zaprezentowane przy okazji opisu zdarzeń.

### Przykład implementacji

Na koniec przedstawimy kompletny przykład użycia timera, który jest zarządzany z poziomu aplikacji Remote.



#### Ustawianie godziny i dni timera z poziomu aplikacji DOMIQ/Remote.

##### Interfejs użytkownika (zakładka Remote):

- Dodaj element typu **Czas**.
- Kliknij na nim dwukrotnie i uzupełnij jego właściwości:
  - W polu **Opis** wpisz opis kontrolki.
  - W komórce **Kanał dla godzin** wpisz *MEM.godzina*.
  - W polu **Kanał dla minut** wpisz *MEM.minuta*.
  - Nazwy tu przyjęte są umowne i możesz wprowadzić inne.
- Dodaj element typu **Wybór**.
  - Kliknij na nim dwukrotnie i uzupełnij jego właściwości:
    - W polu **Opis** wpisz opis, np. *Dni tygodnia*.
    - W komórce **Kanał** wpisz nazwę zmiennej, w której będą zapamiętywane dni tygodnia, kiedy zdarzenie ma zostać wywołane, np. *MEM.dni*.
    - Zaznaczając pole wyboru **Wielokrotny wybór** użytkownik ma możliwość wyboru opcji, kiedy zdarzenie ma być wywoływane.
    - Po zaznaczeniu pola wyboru **Przynajmniej jeden** niemożliwe będzie odznaczenie wszystkich dni tygodnia w menu aplikacji. Użytkownik będzie musiał zaznaczyć co najmniej jedną opcję, kiedy timer będzie wywoływany.
- Zapisz zakładkę **Remote**, aby zatwierdzić ustawienia.

##### Definicja timera:

- Dodaj nowy timer i uzupełnij jego właściwości:
- W komórce **Opis** wpisz jego opis w zależności od przeznaczenia timera.
- Uzupełnij atrybuty w sekcji **Szczegóły** - tu wykonujemy odwołanie do nazw zmiennych MEM, które zostały zadeklarowane w interfejsie Remote.
  - W polu **Godzina** wpisz: *MEM.godzina*
  - W polu **Minuta** wpisz: *MEM.minuta*
  - W polu **Dzień tygodnia** wpisz: *MEM.dni*
- W sekcji **Akcje** zdefiniuj działania, które będą wykonane po wywołaniu timera.
- Zapisz zakładkę **Timery**, aby zatwierdzić ustawienia.

## 8.4. Logika

Moduł Base umożliwia tworzenie i uruchamianie własnych skryptów w języku Lua. Jest to zdecydowanie najbardziej zaawansowana funkcjonalność w module Base, która wymaga wiedzy z zakresu programowania. Użycie skryptów nie jest obligatoryjne i zdecydowana większość funkcjonalności w systemie DOMIQ może być skonfigurowana bez ich użycia.

**Sam opis języka Lua i jego standardowych modułów nie jest ujęty w niniejszej instrukcji.**

Zakładka **Logika** zawiera wbudowany edytor skryptów Lua, z funkcją kolorowania składni oraz autouzupełnianiem wprowadzanego kodu. Możesz użyć także dowolnego innego edytora kodu źródłowego, a następnie po prostu wkleić kod do zakładki **Logika**.

Pod oknem edytora znajduje się okno konsoli, w którym wyświetlane są logi. W logach znajdziesz informacje o ewentualnych błędach napotkanych przy wykonywaniu skryptu (nr linii wraz z opisem błędu), a także są tam wyświetlane wpisy będące efektem wywołania funkcji `print`.

Nad edytorem skryptów znajduje się pasek z przyciskami:

- **Wyczyść** - Kliknij, aby wyczyścić dane konsoli.
- **Kopiuj** - Kopiuje do schowka zawartość konsoli.
- **Zatrzymaj** - Jeżeli opcja jest zaznaczona, wówczas wpisy w konsoli nie przewijają się automatycznie.
- **Zapisz** - Kliknij, aby zapisać skrypt. Zapisanie kodu powoduje także jego przeładowanie i wykonanie. Zatem wszystkie deklaracje zmiennych, a także linie zawierające wywołania funkcji zostaną wykonane.



Założmy, że w edytorze znajduje się następujący kod:

```
function test_fun()
    print('To jest funkcja testowa')
end
```

```
test_fun()
```

Każdorazowe zapisanie zakładki spowoduje wykonanie linii `test_fun()`, będącej wywołaniem funkcji `test_fun`. Efektem będzie wypisanie w konsoli: `To jest funkcja testowa`.

Prawidłowe przeładowanie skryptu zostaje każdorazowo potwierdzone następującym wpisem w konsoli:  
INFO: SCRIPT RELOADED AT <godzina\_wykonania>

- **Przywóć** - Kliknij, aby przywrócić ostatnio zapisaną wersję skryptu.

### Uwagi ogólne

Zapoznaj się z ogólnymi zasadami wykonywania kodu dotyczącymi pisania i wykonania kodu w zakładce **Logika**:

- zmienne zadeklarowane poza jakąkolwiek funkcją są globalne dla całego kodu w zakładce **Logika**.
- nie należy definiować funkcji jako funkcji lokalnych z użyciem słowa `local`.
- funkcje i zmienne zawarte w skryptach zaimportowanych z użyciem funkcji `import` są globalne dla reszty kodu w zakładce **Logika**.
- każdorazowe zapisanie zakładki **Logika** powoduje skompilowanie kodu i uruchomienie wywołań funkcji, które znajdują się w kodzie.

## 8. Automatyzacja

### Moduły i funkcje standardowe

Na potrzeby edytora **Logika** język Lua pracuje w limitowanym środowisku. Oznacza to, że nie wszystkie moduły języka są dostępne do użycia. Jest to funkcja bezpieczeństwa uniemożliwiająca uruchomienie szkodliwego kodu.

Poniżej znajdziesz listę standardowych modułów i funkcji języka Lua, które są dostępne do użycia.

- **Moduły**
  - `table`
  - `string`
  - `math`
- **Funkcje:**
  - `print`
  - `time` - funkcja z modułu `os`, wywołanie bez przedrostka `os`.
  - `pairs`
  - `ipairs`
  - `assert`
  - `date` - funkcja z modułu `os`; wywołanie bez przedrostka `os`.
  - `type`
  - `pcall`
  - `tostring`
  - `tonumber`
  - `unpack`
  - `setmetatable`
  - `getmetatable`

### Moduły i funkcje niestandardowe

Moduły i funkcje niestandardowe są naszą własną implementacją, stanowiącą rozszerzenie możliwości języka Lua.

#### 1. Moduły

##### **timer**

Moduł `timer` umożliwia tworzenie i uruchamianie asynchronicznych timerów. Timer asynchroniczny to taki, który wykona przypisaną do niego funkcjonalność po upływie określonego czasu bez blokowania wykonywania kodu źródłowego. Moduł udostępnia dwa rodzaj timerów:

- timery jednorazowe
- timery cykliczne

##### **Timery jednorazowe**

Timer jednorazowy wywołuje przypisaną do niego funkcję po upływie określonego czasu.

##### 1. Tworzenie instancji timera

Wywołanie funkcji `delay` z modułu `timer` zwraca nową instancję timera (obiekt): `zmienna = timer.delay(opóźnienie)`

Opóźnienie wyrażane jest w sekundach i może być przekazane jako liczba lub jako zmienna, która zawiera liczbę. Możliwe jest również użycie wartości ułamkowych.

Przykład użycia: `local some_timer = timer.delay(10)` lub

`local some_timer = timer.delay(some_var)`, gdzie `some_var` przechowuje wartość liczbową.

Na tym etapie timer został utworzony, ale jest w stanie wstrzymania - odliczanie jeszcze się nie rozpoczęło.

Timer należy przypisać do zmiennej, która jest zmienną globalną z punktu widzenia skryptu. Przystuduj poniższe przykłady:



#### Prawidłowa deklaracja :

```
local tmr -- zadeklarowanie zmiennej na poziomie globalnym dla
całego skryptu w zakładce Logika

function some_function()
    tmr = timer.delay(30)
    tmr:on('timeout', function()
        -- kod do wykonania
    end)
    tmr:start()
end
```

#### Nieprawidłowa deklaracja:

```
function some_function()
    local tmr = timer.delay(30)
    tmr:on('timeout', function()
        -- kod do wykonania
    end)
    tmr:start()
end
```

W przykładzie powyżej, zmienna `tmr` jest lokalna wewnątrz funkcji `some_function`, zatem kiedy funkcja skończy wykonywanie, zmienna `tmr` przestanie istnieć.

## 2. Dostępne metody:

Instancja timera udostępnia następujące metody:

- `on('timeout', funkcja_do_wykonania)`  
To metoda, która pozwala przypisać funkcję, która zostanie wykonana po upływie odliczania.



```
instancja_timera:on('timeout', function()
    print("Odliczanie zakończone")
    -- tu wprowadź kod, który będzie wykonany po zakończeniu
    odliczania
end)
```

Wewnątrz funkcji przypisanej do timera może znajdować się dowolny kod, np. wywołanie innej funkcji, wysłanie komendy do urządzenia w systemie DOMIQ, wysłanie żądania do zewnętrznego serwera itd. Funkcja, która zostaje przekazana jako argument do metody `on` ma dostęp do wszystkich zmiennych i funkcji znajdujących się w otaczającym środowisku i do środowiska globalnego.

- `start()` - Metoda służąca do uruchomienia odliczania.
- `stop()` - Metoda służąca do zatrzymania odliczania. Timer zatrzymany metodą `stop` może być ponownie uruchomiony z użyciem metody `again`.
- `cancel()` - Anuluje odliczanie bez możliwości ponownego jego uruchomienia. Po wywołaniu tej metody, obiekt timera znajduje się w stanie nieaktywności i należy go zainicjować ponownie przez utworzenie nowej instancji.

## 8. Automatyzacja

- `again()` - Rozpoczyna odliczanie od nowa. Tę funkcję można wywołać w timerze, który został zatrzymany przy użyciu metody `stop`. Wówczas timer rozpoczyna odliczanie od początku. Gdy metoda `again` zostanie wywołana w timerze, którego odliczanie już płynie, wówczas licznik odliczania zostanie zresetowany i odliczanie rozpocznie się od nowa. Wykorzystując metodę `again` możliwe jest utworzenie timera, który będzie wywoływał się automatycznie ponownie, po wykonaniu przypisanych do niego zadań.



Przykłady wywołania poszczególnych metod:

```
instancja_timera:start() - uruchomienie odliczania;
```

```
instancja_timera:stop() - zastopowanie odliczania;
```

```
instancja_timera:cancel() - anulowanie timera, po nim nie można ponownie uruchomić odliczania;
```

```
instancja_timera:again() - reset odliczania;
```

Automatyczne wznowienie timera:

```
local tmr = timer.delay(10)
tmr:on('timeout', function()
    -- zadanie 1
    -- zadanie 2
    -- zadanie n
    tmr:again() -- automatyczne wznowienie timera
end)
tmr:start()
```

### 3. Usunięcie timera

Jeżeli timer nie jest już dłużej potrzebny w skrypcie należy go usunąć i skasować zawartość zmiennej, do której timer był przypisany. Załóżmy, że timer był przypisany do zmiennej `some_timer`, wówczas należy wywołać następujący kod:

```
some_timer:stop()
some_timer = nil
```

### Timery cykliczne

Timery cykliczne wykonują odliczanie i przypisana do nich funkcję powtarzalnie, do momentu ich zatrzymania.

#### 1. Tworzenie instancji timera

Obowiązują tu te same zasady, jak dla timerów jednorazowych. Do tworzenia timerów cyklicznych służy metoda `periodic`.

```
zmienna = timer.periodic(opóźnienie)
```

Przykład użycia: `local some_timer = timer.periodic(10)` lub

`local some_timer = timer.periodic(some_var)`, gdzie `some_var` przechowuje wartość liczbową.

Dostępne metody są identyczne, jak dla timerów jednorazowych poza metodą `again`, która jest niedostępna.

## math

Dostępne są 3 dodatkowe funkcje, które dodaliśmy do standardowego modułu `math`.

### 1. `sum`

Funkcja, która umożliwia obliczenie sumy dowolnej liczby elementów. Argumenty należy przekazać w rozdzielone przecinkami.



```
local suma = sum(1,3,5,7)
print(suma) -- 16
```

### 2. `avg`

Funkcja do obliczania średniej arytmetycznej z przekazanego zbioru liczb.



```
local srednia = avg(1,3,5,7)
print(srednia) -- 4
```

### 3. `round`

Funkcja jako wynik zwraca liczbę zaokrągloną do najbliższej liczby całkowitej.



```
local calkowita = round(16.052)
print(calkowita) -- 16
local calkowita = round(16.521)
print(calkowita) -- 17
```

## tcp

Moduł `tcp` umożliwia tworzenie asynchronicznych gniazd TCP (ang. *async socket*). Gniazdo asynchroniczne to takie, które nie blokuje pozostałego kodu w skrypcie, oczekując na odebranie lub wysłanie danych.

Gniazdo TCP może pracować, jako klient lub jako serwer. Dzięki temu modułowi z łatwością utworzysz własne integracje z zewnętrznym oprogramowaniem np. wysyłanie żądań do zewnętrznych serwerów oraz obsługę odpowiedzi. Moduł `tcp` jest powszechnie używany w systemie operacyjnym modułu **Base**.

### 1. Tworzenie gniazda (socketu)

Aby utworzyć nowy socket należy wywołać funkcję `socket` z modułu `tcp`:

```
local sock = tcp.socket().
```

Do zmiennej `sock` została przypisana nowa instancja socketu.

### 2. Dostępne metody.

Instancja socketu umożliwia wywołanie metod opisanych poniżej. We wszystkich opisach zakładamy, że instancja gniazda przypisana jest do zmiennej `sock`.

- `connect(host, port)` - Metoda służy do nawiązania połączenia ze wskazanym hostem, na wskazanym porcie.
 

```
sock:connect('192.168.1.100', 80)
sock:connect('example.com', 80)
```
- `mode(mode)` - Metoda służy do wyboru trybu pracy. Dostępne są następujące tryby pracy:
  - `binary` - Odebranie porcji danych emituje zdarzenie `'data'`, które z kolei wywołuje przypisaną funkcję obsługującą (handler), przekazując do niej odebrane dane. Jest to domyślny tryb pracy socketu.
 

Sposób użycia: `sock:mode('binary')`

## 8. Automatyzacja

- `line` - Tryb liniowy. Socket odbierając znak `\r` (CR) lub `\n` (LF) emituje zdarzenie `'line'`, które z kolei wywołuje funkcję obsługującą (handler), przekazując do niej odebrane dane.  
Sposób użycia: `sock:mode('line')`
- `send(data)` - Metoda służąca do wysyłania danych przez socket.  
`sock:send(data)`
- `on(event, handler)` - Metoda służąca do przypisywania funkcji obsługujących (handlerów) do zdarzeń, które są emitowane przez socket. Funkcje obsługujące można zdefiniować wewnątrz metody `on` - jako tzw. funkcja anonimowa lub przekazać nazwę funkcji, która została zdefiniowana w innym miejscu kodu. Przystudiuj dwa przykłady:
  1. Funkcja anonimowa jako handler.  

```
sock:on('connect', function(client, addr, port)
  -- kod do wykonania
end)
```
  2. Funkcja zdefiniowana w innym miejscu kodu:  

```
function handle_connect(client, addr, port)
  -- kod do wykonania
end

sock:on('connect', handle_connect)
```

Socket może wyemitować następujące rodzaje zdarzeń:

- `'connect'` - zdarzenie jest emitowane w dwóch przypadkach:
  - Jako efekt wywołania metody `connect`, gdy udało się nawiązać połączenie ze wskazanym hostem - socket pełni tu rolę klienta.  

```
sock:on('connect', function()
  print('Socket Connected')
  -- reszta kodu do wykonania po nazwiązaniu połączenia
end)
```
  - Jako reakcja na połączenie przychodzące - socket pełni tu rolę serwera.  

```
function handle_connect(client, addr, port)
  -- kod do wykonania
end

sock:on('connect', handle_connect)
```

Funkcja obsługująca jako argumenty otrzymuje socket oraz adres i port identyfikujący połączenie przychodzące.

**WAŻNE! Wszystkie poniższe zdarzenia należy obsługiwać wewnątrz funkcji przypisanej do obsługi zdarzenia `'connect'`. Dzięki temu funkcje zostaną zadeklarowane i przypisane wyłącznie, gdy połączenie faktycznie dojdzie do skutku.**

- `'data'` - zdarzenie jest emitowane po odebraniu porcji danych, gdy socket pracuje w trybie `binary`. Przypisany handler otrzymuje jako argument odebrane dane.  

```
sock:on('data', function(data)
  print('RECV DATA')
  -- kod przetwarzający odebrane dane
end)
```
- `'line'` - zdarzenie jest emitowane po odebraniu porcji danych, gdy socket pracuje w trybie `line`. Przypisany handler otrzymuje jako argument odebrane dane.  

```
sock:on('line', function(data)
  print('RECV LINE', data)
  -- kod przetwarzający odebrane dane
end)
```
- `'drain'` - zdarzenie jest emitowane, gdy bufor nadawczy socketu został opróżniony. Dane trafiają do bufora nadawczego, gdy nie mogły być natychmiastowo wysłane po wywołaniu metody



`send`. To zdarzenie jest szczególnie przydatne, gdy wysyłasz dane porcjami i wysłanie kolejnej porcji powinno nastąpić tylko wtedy, gdy poprzednia porcja została dostarczona.

- `'close'` - zdarzenie emitowane w przypadku zamknięcia połączenia. Jest ono emitowane niezależnie od tego, która ze stron je zamknieła. Przypisana funkcja obsługująca otrzymuje jako argument komunikat o ewentualnych błędach.

```
sock:on('close', function(msg)
    print("Socket Closed: " .. msg)
end)
```

To zdarzenie można użyć np. do implementacji autowznowienia połączenia w przypadku jego zamknięcia.

- `'error'` - zdarzenie emitowane w przypadku błędu, który wystąpił w połączeniu. Połączenie zostanie automatycznie zamknięte po wystąpieniu błędu. Przypisana funkcja obsługująca otrzymuje jako argument komunikat o błędzie.

```
sock:on('error', function(err)
    print("Socket Error: " .. err)
end)
```

- `off(event)` - Metoda będąca przeciwieństwem do metody `on`. Umożliwia anulowanie działanie funkcji obsługującej przypisanej do danego zdarzenia. Jest to rzadko używana funkcjonalność, jednakże czasem bardzo przydatna. Jednym ze scenariuszy jest potrzeba natychmiastowego zaprzestania przetwarzania danych, zanim połączenie zostanie ostatecznie zamknięte. Wówczas wywołanie metody `off` jest właściwym rozwiązaniem.



```
sock:off('data')
```

Socket od tego momentu nie będzie rozgłaszał zdarzenia typu `'data'`.

- `close()` - Metoda służąca do zamykania połączenia.

```
sock:close()
```

Jeżeli połączenie nie jest już dłużej potrzebne, zaleca się jego zamknięcie (wywołanie `close()`) i skasowanie zawartości zmiennej, do której socket był przypisany (przypisanie do zmiennej wartości `nil`).

- `bind(port)` - Metoda przypisująca numer portu, na którym socket będzie oczekiwał przychodzących połączeń. Nr portu można przekazać jako liczbę lub jako zmienną, która zawiera liczbę.

```
sock:bind(1234)
```

lub

```
local port = 1234
```

```
sock:bind(port)
```

- `listen()` - Metoda uruchamiająca nasłuch na przychodzące połączenia, na porcie wskazanym metodą `bind`.

### 3. Przykłady użycia

Na koniec przedstawiamy dwa przykłady: pierwszy z nich pokazuje użycie socketu jako połączenia typu klient, a drugi użycie socketu TCP do implementacji funkcji serwerowych.



### Socket jako klient.

Przykład przedstawia nawiązanie połączenia z serwerem o adresie `example.com`, wysłanie żądania HTTP oraz obsługę odpowiedzi z serwera.

```
local socket
local sendRequest

function createConnection(req)
    socket = tcp.socket()
    socket:connect("example.com",80)
    socket:on('connect', function()
        tr:print('Example Server Connected')
        socket:mode('line')
        sendRequest(req)
        socket:on("close", function()
            socket = nil
        end)
        socket:on('line', function(data)
            print(data)
            -- the rest of handler's code
        end)
    end)
end

sendRequest = function(req)
    if socket ~= nil then
        local qm = string.find(req,"?") -- check for a query mark
        local name,query
        if qm then
            name,query = string.match(req,"([^?]+)%?([^?]+)" -- if
query mark present, split request string on it
        else
            name = req
        end

        local path = "http://example.com/somepath/"..name)
        if query then
            socket:send(string.format("GET %s%s HTTP/
1.1\r\nHost:%s\r\nConnection: keep-alive\r\n\r\n",path, query, "example.com"))
        else
            socket:send(string.format("GET %s HTTP/
1.1\r\nHost:%s\r\nConnection: keep-alive\r\n\r\n",path,"example.com"))
        end

    else
        createConnection(req)
    end
end

sendRequest('sometestrequest')
```



### Socket jako serwer.

W poniższym przykładzie socket będzie nasłuchiwał na połączenia przychodzące na porcie 1234. W przypadku nawiązania połączenia będzie odsyłał wszystkie odebrane dane (echo serwer).

```

local sock
local quiet = false

local function handle_connect(client, addr, port)
    local buf
    client:send('I WILL ECHO EVERYTHING!\n')
    if not quiet then
        print('CONNECT', addr, port)
    end
    client:mode('line')

    client:on('line', function(buf)
        print('RX', buf)
        client:send(buf..'\\n')
    end)

    client:on('end', function()
        if not quiet then
            print('DISCONNECT')
        end
    end)

    client:on('error', function(err)
        print('ERROR', client, err)
    end)
end

local function handle_error(err)
    print('ERROR', sock, err)
end

sock = tcp.socket()
sock:bind(1234)
sock:on('connect', handle_connect)
sock:on('error', handle_error)
sock:listen()
print("Listening", sock)

```

## 8. Automatyzacja

### udp

Moduł `udp` umożliwia tworzenie asynchronicznych gniazd UDP (ang. `async socket`). Gniazdo asynchroniczne to takie, które nie blokuje pozostałego kodu w skrypcie, oczekując na odebranie lub wysłanie danych. W swojej strukturze moduł `udp` jest bardzo podobny do modułu `tcp`. Różnice między tymi dwoma modułami wynikają z różnic pomiędzy protokołem TCP a UDP - protokół UDP jest protokołem bezstanowym. Zatem nie znajdziemy tu metody `connect`, jak i zdarzeń dla obsługi połączeń przychodzących.

Gniazdo UDP może pracować, jako klient lub jako serwer. Dzięki temu modułowi z łatwością utworzysz własne integracje z zewnętrznym oprogramowaniem np. wysyłanie żądań do zewnętrznych serwerów oraz obsługę odpowiedzi.

#### 1. Tworzenie gniazda (socketu)

Aby utworzyć nowy socket należy wywołać funkcję `socket` z modułu `udp`:

```
local sock = udp.socket().
```

Do zmiennej `sock` została przypisana nowa instancja socketu.

#### 2. Dostępne metody.

Instancja socketu umożliwia wywołanie metod opisanych poniżej. We wszystkich opisach zakładamy, że instancja gniazda przypisana jest do zmiennej `sock`.

- `send(data)` - Metoda służąca do wysyłania danych przez socket.  
`sock:send(data)`
- `on(event, handler)` - Metoda służąca do przypisywania funkcji obsługujących (handlerów) do zdarzeń, które są emitowane przez socket. Funkcje obsługujące można zdefiniować wewnątrz metody `on` - jako tzw. funkcja anonimowa lub przekazać nazwę funkcji, która została zdefiniowana w innym miejscu kodu. Przystudiuj dwa przykłady:

##### 1. Funkcja anonimowa jako handler.

```
sock:on('message', function(data, addr, port)
    -- kod do wykonania
end)
```

##### 2. Funkcja zdefiniowana w innym miejscu kodu:

```
function handle_message(data, addr, port)
    -- kod do wykonania
end
```

```
sock:on('message', handle_message)
```

Socket może wyemitować następujące rodzaje zdarzeń:

- `'message'` - zdarzenie jest emitowane po odebraniu porcji danych. Przypisana funkcja obsługująca otrzymuje jako argument odebrane dane oraz adres i numer portu nadawcy.

```
sock:on('message', function(data, addr, port)
    print('RECV DATA')
    -- kod przetwarzający odebrane dane
end)
```

- `'error'` - zdarzenie emitowane w przypadku błędu. Przypisana funkcja obsługująca otrzymuje jako argument komunikat o błędzie.

```
sock:on('error', function(err)
    print("Socket Error: "..err)
end)
```

- `off(event)` - Metoda będąca przeciwieństwem do metody `on`. Umożliwia anulowanie działanie funkcji obsługującej przypisaną do danego zdarzenia.



```
sock:off('message')
```

Socket od tego momentu nie będzie rozgłaszał zdarzenia typu 'message'.

- `bind(port)` - Metoda przypisująca numer portu, na którym socket będzie oczekiwał przychodzących danych. Nr portu można przekazać jako liczbę lub jako zmienną, która zawiera liczbę.  

```
sock:bind(1234)
```

 lub  

```
local port = 1234
sock:bind(port)
```
- `addMembership(adres)` - Metoda umożliwiająca przypisanie do grupy multicast.
- `dropMembership(adres)` - Metoda umożliwiająca porzucenie przynależności do grupy multicast

### 3. Przykłady użycia



#### Socket jako serwer.

W poniższym przykładzie, socket będzie nasłuchiwał na porcie 1234. W przypadku odebrania danych, będzie odsyłał te same dane (echo serwer).

```
sock = udp.socket()
sock:bind(1234)
sock:on('message', function(data, addr, port)
    print('RX', addr, port)
    print(data..'\n')
    sock:send(data, addr, port)
end)

sock:on('error', function(err)
    print('ERROR', sock, err)
end)

print("Listening", sock)
```

#### Socket jako klient

Przykład pokazuje, jak wysyłać dane do dowolnego hosta.

```
sock = udp.socket()
sock:on('message', function(data, addr, port)
    print('RX', addr, port)
    print(data..'\n')
    -- the rest of handler's code
end)

sock:on('error', function(err)
    print('ERROR', sock, err)
end)

function sendData(data, addr, port)
    sock:send(data, addr, port)
end

sendData("some data", '192.168.1.100', 44544)
```

## 8. Automatyzacja

### webapi

Moduł umożliwiający z poziomu kodu w zakładce **Logika**, obsługę żądań HTTP przez wbudowany w module Base serwer HTTP. W ten sposób możesz utworzyć własne API HTTP do integracji modułu Base z dowolnym oprogramowaniem. Każdy adres żądania może mieć zdefiniowaną osobną funkcję obsługującą (handler).

- `public(name, handler)`

Metoda `public` służy do rejestrowania funkcji obsługujących dla żądań, które nie wymagają autoryzacji.

`name` - nazwa identyfikująca żądanie

`handler` - nazwa funkcji wywoływanej do obsługi żądania.

Żądanie musi zawsze mieć określoną strukturę wg poniższego wzoru:

`http://<host>/call/public?vm=LOGIC&call=<nazwa_żądania>&<dowolne zmienne>`

Przestuduj poniższy przykład:



```
http://192.168.1.30/call/public?
vm=LOGIC&call=examplerequest&message=somemessage&value=100
```

Powyższe żądanie zostanie odebrane z nazwą `examplerequest` oraz dwoma dodatkowymi parametrami: `message` z wartością `somemessage` oraz `value` z wartością `100`.

Kod w zakładce **Logika**:

```
function handleExampleRequest(params)
  -- params to tablica parametrów przekazanych w żądaniu
  print('message: '..params.message)
  print('value: '..params.value)
end
```

```
webapi.public('examplerequest', handleExampleRequest)
```

Powyższe wywołanie `webapi.public` zarejestrowało żądanie pod nazwą `examplerequest` i przypisało do niego funkcję obsługującą `handleExampleRequest`.

Funkcja obsługująca w momencie wywołania otrzymuje jako argument, tablicę parametrów, które zostały użyte w żądaniu. Jeżeli w żądaniu nie było dodatkowych parametrów, wówczas funkcja otrzyma pustą tablicę.

2. `secure(name, handler)`

Metoda `secure` działa analogicznie, jak metoda `public`, jednakże wymaga aby żądanie było autoryzowane z użyciem loginu i hasła do konfiguratora. Dane autoryzacji należy przesłać w nagłówku `Authorization`. Dane mogą być zakodowane metodą `Basic` lub `Digest`.

Użycie metody `secure` różni się jedynie fragmentem żądania – zamiast `call/public` trzeba użyć `call/secure`. Oraz należy wywołać `webapi.secure` zamiast `webapi.public`.

Nawiązując do przykładu dla metody `public`, mamy wówczas:

```
http://192.168.1.30/call/
```

```
secure?vm=LOGIC&call=examplerequest&message=somemessage&value=100
oraz
```

```
webapi.secure('examplerequest', handleExampleRequest)
```

## 2. Funkcje

### use

Funkcja służy do deklarowania obiektów, które pozwalają na interakcję ze wskazanym identyfikatorem w stanie modułu Base. Utworzone w ten sposób obiekty można używać w całej zakładce **Logika**, przekazując je między funkcjami lub używać ich jako zmiennych globalnych.

#### 1. Deklaracja obiektu

Ogólna składnia jest następująca:

```
local somevariable = use 'identyfikator', gdzie identyfikator należy zastąpić konkretnym identyfikatorem ze stanu.
```

```
local somevariable = use 'LCN.output.0.10.1'
```

**WAŻNE!** Obiekt tworzy automatycznie subskrypcję na odbieranie zdarzeń o zmianie stanu wskazanego identyfikatora. Umożliwia to programową obsługę zdarzeń - patrz metoda `onchange`.

#### 2. Własności obiektu

- `value` - Aktualna wartość identyfikatora przypisanego do obiektu.
- `new` - Nowa wartość identyfikatora odebrana przy użyciu metody `onchange`.
- `old` - Ostatnia znana wartość identyfikatora. Własność ustawiana jest przy wywołaniu metody `onchange`. Ułatwia porównywanie bieżącej i poprzedniej wartości identyfikatora.
- `age` - Wartość w sekundach od czasu ostatniej zmiany stanu.

#### 3. Dostępne metody

Na potrzeby opisu dostępnych metod przyjmujemy obiekt utworzony w punkcie 1 - identyfikator odnoszący się do wyjścia ściemnianego nr 1 w module LCN o adresie 10.

- `set(value)` - Metoda umożliwiająca nadanie nowej wartości identyfikatorowi. Wartość przekazana do funkcji `set` może być dowolna, istotne jest aby była dopasowana do rodzaju identyfikatora.  
Składnia ogólna: `obj:set(value)`



```
somevariable:set(100)
```

Powyższe wywołanie metody `set` pociąga za sobą dwie akcje:

1. Przypisuje w stanie modułu **Base** wartość 100 do identyfikatora `LCN.output.0.10.1`
2. Wysłanie komendy z wartością 100 do wyjścia ściemnianego nr 1 w module LCN.

- `get()` - Metoda zwracająca bieżący stan identyfikatora przypisanego do obiektu.



```
local currentValue = somevariable:get()
print(currentValue) --> 100
```

- `channel()` - Metoda zwraca nazwę identyfikatora przypisanego do obiektu.



```
local name = somevariable:channel()
print(name) --> LCN.output.0.10.1
```

- `onchange(new, old)` - Metoda umożliwia przypisanie funkcji obsługującej (handlera), która będzie wywołana za każdym razem, gdy identyfikator zmieni swój stan. Funkcja obsługująca jako argumenty otrzymuje nowy stan identyfikatora oraz jego wartość sprzed zmiany.

## 8. Automatyzacja



Oba poniższe zapisy są tożsame:

```
function somevariable:onchange(new,old)
    print("New value: "..new)
    print("Old value: "..old)
end

somevariable.onchange = function(somevariable,new,old)
    print("New value: "..new)
    print("Old value: "..old)
end
```

- `ontick()` - Metoda umożliwia przypisanie funkcji obsługującej, która będzie wywoływana co sekundę. Jest funkcjonalność podobna do `timer.periodic`.



Oba poniższe zapisy są tożsame:

```
function somevariable:ontick()
    print("Ticking")
end

somevariable.ontick = function()
    print("Ticking")
end
```

### age

Funkcja zwraca czas wyrażony w sekundach od ostatniej zmiany stanu danego identyfikatora. Funkcja jako argument przyjmuje string z nazwą identyfikatora lub obiekt utworzony z użyciem metody `use`.



Identyfikator jako string

```
local age = age('LCN.output.0.10.1')
print(age) --> 1234
```

Przekazanie obiektu

```
local somevariable = use'LCN.output.0.10.1'
local age = age(somevariable)
print(age) --> 1234
```

### systemtime

Funkcja zwraca czasu od momentu uruchomienia modułu Base. Jest to wygodne narzędzie do pomiaru upływu czasu pomiędzy dwoma zdarzeniami w systemie, gdyż funkcja `systemtime` w odróżnieniu od standardowej funkcji `time`, zwraca czas z dokładnością większą niż pełne sekundy.





Przykład pomiaru czasu pomiędzy dwoma miejscami miejscami kodu

```
function veryResourceHungryFunction()
    -- some very time consuming operations go here
end

local start = systime()
veryResourceHungryFunction()
local end = systime()
print("Duration: "..end - start) --> 8.8125
```

## command

Funkcja umożliwia wysłanie komendy lub rozgłoszenie zdarzenia z określoną wartością. Wymagane jest tu użycie przedrostka **C.** (komendy) lub **E.** (zdarzenia).

Dopuszczalne są dwa rodzaje wywołania funkcji, oba są równoważne, choć pierwsza metoda jest bardziej optymalna pod względem wydajności.

```
command(komenda/zdarzenie,wartość) lub
command(komenda/zdarzenie=wartość)
```



- 1. Włączenie przełącznika np. 1 w module Satel INT-IORS z adresem 3 sterowany .**  
`command('C.EXP.output.3.1,1)`
- 2. Wysterowanie wyjścia ściemnianego nr 2 w module LCN o adresie 10 na 100%.**  
`command('C.LCN.output.0.10.2,100)`
- 3. Rozgłoszenie zdarzenia na przełączenie ekranu wizualizacji w aplikacji Remote.**  
 Na potrzeby przykładu, załóżmy że ekran ma nazwę `test`. Kiedy taka akcja zostanie wykonana, wszystkie aplikacje Remote, które aktualnie wyświetlają wizualizację pokażą ekran o nazwie `test`.  
`command('E.DISPLAY.screen,"test")`

## import

Funkcja służy do importowania skryptów wgranych w zakładce **Zasoby** do kodu w **Logice**. Funkcje i zmienne zaimportowanego skryptu stają się globalne na poziomie całego kodu w zakładce **Logika**.



```
import'script_name'
```

## set

Funkcja umożliwiająca ustawienie stanu dowolnego identyfikatora. Różnica między funkcją `set` a metodą `set` obiektu utworzonego przy użyciu funkcji `use` jest następująca: funkcja `set` ustawia jedynie wartość identyfikatora w tabeli stanu w module Base, nie powodując wysłania komendy do urządzenia, które kryje się za tym identyfikatorem. Natomiast metoda `set` obiektu robi obie czynności. Zatem jeżeli chcesz zmienić identyfikator, który jest jedynie zmienną wirtualną np. zmienną typu VAR, wówczas wystarczy użyć funkcji `set`.

## 8. Automatyzacja



1. Zmiana stanu zmiennej `VAR.test` na wartość 100.

```
set('VAR.test',100)
```

2. Porównaj różnice między funkcją `set`, a metodą `set` obiektu

```
set('LCN.output.0.10.1',100) -- ustawia stan na wartość 100
```

```
local somevariable = use'LCN.output.0.10.1'
```

```
somevariable:set(100) --> ustawia stan i wysyła komendę do modułu LCN
```

### get

Funkcja zwraca aktualny stan wskazanego identyfikatora.



```
local value = get('LCN.output.0.10.1')
```

```
print(value) --> wypisuje aktualną wartość wyjścia ściemnianego
```

### set\_attr

Funkcja służy do zmiany wartości atrybutów dowolnego identyfikatora w stanie modułu Base. Każdy identyfikator ma atrybut `description` (opis) oraz `age` (czas od ostatniej zmiany).

Funkcja przyjmuje trzy argumenty: nazwa identyfikatora, nazwa atrybutu, nowa wartość atrybutu



```
set_attr("LCN.output.0.10.1","description","lampa sufit")
```

### get\_attr

Funkcja zwraca wartość wskazanego atrybutu identyfikatora. Do funkcji należy przekazać nazwę identyfikatora i nazwę atrybutu.



```
local description = get_attr("IDS.input.10","description")
```

# Rozdział 9

# Integracje

Jedną z głównych cech systemu DOMIQ są jego możliwości integracji z zewnętrznymi systemami automatyki lub zewnętrznym oprogramowaniem z użyciem rozmaitych protokołów i standardów komunikacyjnych.

W tym rozdziale zostaną opisane dostępne integracje wraz z interfejsami konfiguracyjnymi do nich przeznaczonymi.

# 9.1. Ekspandery

Moduł DOMIQ/Expander w połączeniu z modułem Base umożliwia podłączanie sterowników Satel INT-IORS lub INT-ORS (zwanymi dalej ekspanderami), wprost do systemu DOMIQ. Możliwe jest podłączenie do 32 ekspanderów Satel, co oznacza stworzenie instalacji wyposażonej w aż 256 wyjść przekaźnikowych i 256 wejść binarnych. Do wejść można podłączyć przyciski dzwonekowe lub dwustanowe, czujki ruchu, kontaktrony i inne urządzenia, które są w stanie wysterować wejścia ekspandera. Do wyjść można podłączyć dowolne odbiorniki np. światła, rolety, obwody gniazdek elektrycznych, itd. Specjalnie zaprojektowane komendy sterujące dla rolet ułatwiają ich integrację w systemie. Więcej o komendach sterujących roletami przeczytasz w rozdziale 12. Identyfikatory

Zakładka **Ekspandery** stanowi interfejs do konfiguracji integracji z ekspanderami Satel.

### Najważniejsze cechy integracji i konfiguracji:

- Konfiguracja polega na tworzeniu powiązań pomiędzy wejściami a wyjściami ekspanderów.
- Wyjścia ekspanderów można grupować, a grupy następnie przypisywać do wejść, co znacznie skraca proces konfiguracji systemu.
- Każde wejście w instalacji można przypisać do dowolnej liczby wyjść i grup.
- Możliwość konfiguracji zarówno w trybie online, jak i offline. Konfigurację można wykonać bez fizycznie podłączonych ekspanderów, a także bez modułu DOMIQ/Expander. Wystarczy sam moduł Base.
- Przyciski testowe dla wejść, które symulują naruszenie wejścia. Dzięki temu możliwe jest sprawdzenie poprawności utworzonych powiązań.
- Przyciski testowe dla wyjść. Dzięki czemu można sprawdzić działanie każdego wyjścia i poprawność okablowania danego obwodu.

Interfejs został podzielony na trzy widoki:

- **Główny**
- **Grupy**
- **Wyjścia**

W górnym prawym rogu okna znajduje się pasek z przyciskami:

- **Skanuj** – Naciśnięcie uruchamia skanowanie magistrali w poszukiwaniu podłączonych ekspanderów. Skanowanie magistrali jest uruchamiane automatycznie przy starcie modułu Base, jeśli jest podłączony moduł DOMIQ/Expander. Skanowanie ręczne należy wykonać w dwóch przypadkach:
  - podłączenie nowych ekspanderów do magistrali;
  - zmiana adresu jednego lub wielu ekspanderów;
- **Odtwórz** – Odtwarza ostatnią zapisaną konfigurację zakładki Ekspandery.
- **Zapisz** – Zapisuje konfigurację zakładki **Ekspandery** (wszystkie widoki). **WAŻNE:** utworzone powiązania nie będą aktywne dopóki nie zostanie zapisana konfiguracja.

## Główny

Widok **Główny** służy do tworzenia powiązań między wejściami ekspandera a wyjściami i grupami wyjść. Podzielony został na dwie tabele. Tabela po lewej przeznaczona jest do konfiguracji właściwości wejść i będzie dalej zwana **tabelą wejść**. Natomiast tabela po prawej stronie, służy do wyboru wyjść, które mają zostać przypisane do danego wejścia i będzie nazywana **tabelą wyjść**.

### 1. Tabela wejść

**Tabela wejść** zawiera następujące kolumny:

- **Ekspander** – w tej kolumnie wybierany jest ekspander, dla którego wejść chcesz utworzyć powiązania. Ekspandery fizycznie dostępne w instalacji oznaczone są na liście kolorem zielonym i mają status "Podłączony". Dodatkowo te ekspandery, których wejścia mają jakiegokolwiek przypisania jest wyróżniony pogrubieniem.

- **Wejście** – W tej kolumnie wybierane jest wejście, dla którego tworzone jest powiązanie. Obok numeru wejścia wyświetlana jest sygnalizacja stanu wejścia. Gdy wejście jest w stanie nieaktywnym, kółko ma kolor szary, a w stanie aktywnym ma kolor zielony.
- **Komenda** – Zawiera listę dostępnych komend. Do wyboru są następujące komendy:
  - **Przełącz** – każde naruszenie wejścia przełącza przypisane przekaźniki.
  - **Włącz** – każde naruszenie wejścia włącza przypisane przekaźniki (ustawienie logicznej 1).
  - **Wyłącz** – każde naruszenie wejścia wyłącza przekaźniki (ustawienie logicznego 0).
  - **Włącz (monostabilny)** – Przypisane przekaźniki zostaną wysterowane tak długo, jak długo wejście będzie naruszone.
  - **Brak komendy (pusta linia)** – Istnieje również możliwość wyboru pustej komendy. Wówczas do przekaźników nie zostanie wysłana żadna komenda, jednakże naruszenie wejścia spowoduje wygenerowanie zdarzenia w module Base, które może stanowić podstawę do wywołania dowolnej sekwencji działań w systemie.
- **Opis** – Pomocniczy opis wejścia. Wprowadzony opis będzie widoczny również w zakładce **Stan**.
- **Grupy** – Ta kolumna służy do wyboru grup wyjść, które mają zostać przypisane do wejścia. Domyślnie wejścia nie mają przypisanych żadnych grup, co jest sygnalizowane napisem **Przypisz grupy**. Kliknięcie na nim otworzy okno wyboru grup. W oknie wyboru grup zebrane są wszystkie utworzone grupy wyjść wraz z opisami (tworzenie grup zostało opisane w dalszej części tego rozdziału). Po wybraniu grupy/grup, napis **Przypisz grupy** zostanie zastąpiony listą z numerami grup, które zostały wybrane. Aby edytować przypisane grupy, kliknij na liście numerów.
- Przycisk **test** - Umożliwiają testowanie utworzonego powiązania. Naciśnięcie przycisku symuluje naruszenie wejścia. Symulowane w ten sposób naruszenie wejścia może być nieznacznie wolniej wykrywane przez system, niż realne naruszenie.

## 2. Tabela wyjść

Tabela służy do wyboru wyjść, które mają zostać powiązane z wejściami. Podzielona jest na dwie kolumny:

- **Ekspander** – Lista z adresami ekspanderów.
- **Wyjścia** – W tej kolumnie należy wybrać wyjścia, które mają być przypisane do danego wejścia ekspandera. Do wejścia może być przypisana dowolna liczba wyjść z dowolnej liczby ekspanderów. Aby usunąć przypisanie danego wyjścia należy je odznaczyć, a następnie zapisać konfigurację.

## Grupy

Widok **Grupy** przeznaczony jest do tworzenia grup wyjść. Maksymalnie można utworzyć 256 grup (ID od 0 do 255). Do każdej grupy można przypisać dowolną liczbę wyjść w dowolnej liczbie ekspanderów. Widok został podzielony na dwie tabele: **Tabelę grup** oraz **Tabelę wyjść**. Tabela wyjść służy do wyboru wyjść, które mają zostać przypisane do grupy. Tabela ma strukturę i mechanizm działania identyczny, jak w przypadku widoku **Głównego**, dlatego nie będziemy jej opisywać.

### 1. Tabela grup

Tabela służy do tworzenia grup. Procedura dodawania grupy jest następująca:

1. Kliknij przycisk **Dodaj**.
2. W oknie, które zostanie wyświetlone, w polu **Numer** wpisz numer tworzonej grupy (w zakresie od 0 do 255). Podanie numeru grupy jest wymagane. W polu Opis można wprowadzić pomocniczy komentarz dla grupy.
3. Gotowe.

Następnie zaznacz grupę w tabeli i utwórz przypisania wyjść korzystając z **Tabeli wyjść**.

## 9. Integracje

### Wyjścia

Widok **Wyjścia** umożliwia przetestowanie działania przekaźników oraz nadanie im opisów. Tabela ma cztery kolumny:

- **Ekspander** – Lista z adresami ekspanderów.
- **Wyjście** – W tej kolumnie wybierane jest wyjście. Obok numeru wyjścia wyświetlana jest sygnalizacja stanu wejścia. Gdy wejście jest w stanie nieaktywnym, kółko ma kolor szary, a w stanie aktywnym ma kolor zielony.
- **Opis** – W tej kolumnie można wpisać opis dla każdego wyjścia. Opis zostanie zapisany w pliku konfiguracyjnym i będzie również widoczny w zakładce **Stan**.
- Przycisk **test** - Umożliwiają testowanie działania przekaźnika. Każde naciśnięcie przycisku przełącza stan przekaźnika.

### Sterowanie

Sterowanie ekspanderami odbywa się na podstawie utworzonych powiązań – naruszenie wejścia powoduje wystereowanie wyjść i/lub grup wedle zaprogramowanych reguł.

Ponadto ekspandery mogą być sterowane na wszystkie inne sposoby dostępne w systemie DOMIQ, tj. z poziomu wizualizacji, aplikacji Remote, poprzez zdarzenia, reguły czasowe, itd. Moduł Base może reagować zdarzeniami zarówno na zmiany wejść, jak i wyjść. Oprócz powyższych metod, ekspandery mogą być sterowane komendami przy użyciu identyfikatorów. Więcej o identyfikatorach przeczytasz w rozdziale 12.

Identyfikatory

Stan wszystkich wejść i wyjść (których stan jest niezerowy) jest przechowywany w tabeli stanu w module Base, co umożliwia ich zwizualizowanie w systemie DOMIQ.

Dzięki możliwościom integracyjnym systemu DOMIQ, ekspandery mogą być również sterowane na podstawie informacji z innych podsystemów inteligentnego domu, np. w reakcji na uzbrojenie alarmu, zmiany temperatury, itp.

## 9.2. MODBUS

System DOMIQ umożliwia dwukierunkową komunikację z innymi urządzeniami przy użyciu protokołu MODBUS. Powszechność zastosowania tego protokołu sprawia, że otwiera to drogę do integracji bardzo szerokiego spektrum urządzeń w systemie DOMIQ. Integracja z użyciem protokołu MODBUS znajduje zastosowanie m.in w:

- centralach wentylacyjnych,
- klimatyzatorach,
- inwerterach fotowoltycznych,
- licznikach energii elektrycznych,
- czujnikach pomiarowych (m.in. temperatury, wilgotności, VOC),
- stacjach pogodowych,
- sterownikach PLC,
- i wielu innych,

Moduł Base obsługuje trzy wersje protokołu: MODBUS TCP, MODBUS UDP oraz MODBUS RTU. W tej ostatniej odmianie wymagane jest użycie bramki sprzęgającej DOMIQ/Serial-4MB.

We wszystkich rodzajach integracji moduł Base pełni rolę urządzenia typu `master`. Natomiast integrowane urządzenia działają jako `slave`.

### Etapy integracji

1. Pozyskanie dokumentacji integrowanego urządzenia wraz opisami parametrów transmisji (tylko MODBUS RTU) oraz adresacją rejestrów MODBUS. Jest to etap absolutnie konieczny, aby nawiązać komunikację i prawidłowo odczytywać z niego parametry.
2. (Tylko MODBUS RTU): Jeżeli dane urządzenie jest kolejnym urządzeniem, które jest podłączane do magistrali MODBUS zarządzanej przez moduł Base, wówczas należy ustawić parametry transmisji: prędkość i format ramki taki, z jakiego korzystają pozostałe urządzenia.
3. Ustawienie adresu MODBUS integrowanego urządzenia. Zakres adresów wg standardu MODBUS to: 1-247. Urządzenia muszą mieć unikalne adresy w obrębie magistrali.
4. Po stronie konfiguratora modułu Base:
  - Dodanie połączenia i skonfigurowanie jego parametrów.
  - Dodanie urządzenia do połączenia i skonfigurowanie jego parametrów.
  - Skonfigurowanie odczytów rejestrów w obrębie urządzenia.  
**WAŻNE:** Dla celów diagnostyki komunikacji z danym urządzeniem, zalecamy początkowo dodanie pojedynczego rejestru, którego wartość możesz bezbłędnie ustalić, np. większość urządzeń udostępnia rejestry zawierające aktualną datę. Daje to możliwość weryfikacji, czy konfiguracja parametrów tego połączenia i urządzenia jest prawidłowa. W przeciwnym wypadku odczyty rejestrów będą miały nieprawidłowe dane lub ich odczyt nie będzie wcale możliwy.

Konfiguracja po stronie modułu Base będzie opisana w dalszej części tego rozdziału.

## 9. Integracje

### Interfejs konfiguracyjny

Interfejs w zakładce **Modbus** składa się z następujących elementów:

1. Pasek przycisków
2. Struktura
3. Szczegóły
4. Rejestry (tylko, gdy w Strukturze wybrano element typu urządzenie).

#### Pasek przycisków

- **+ Połączenie** - Kliknij, aby dodać nowe połączenie.
- **+ Urządzenie** - Kliknij, aby dodać nowe urządzenie do wybranego połączenia.
- **Importuj urządzenie** - Umożliwia zaimportowanie pliku z konfiguracją urządzenia z innego modułu Base.
- **Eksportuj urządzenie** - Umożliwia zapisanie na dysku komputera konfiguracji wybranego urządzenia.
- **Usuń** - Kliknij, aby usunąć zaznaczony element struktury.

#### Struktura

W oknie **Struktura** wyświetlane są utworzone połączenia wraz z dodanymi do nich urządzeniami. Dla łatwiejszej nawigacji etykiety zawierają najistotniejsze informacje o danym połączeniu/urządzeniu.

Aby usunąć element ze struktury, zaznacz pole wyboru i kliknij **Usuń**.

#### Szczegóły

Formularz **Szczegóły** służy do konfigurowania parametrów poszczególnych połączeń i urządzeń. Poszczególne pola i nastawy zostały omówione w dalszej części tego rozdziału.

#### Rejestry

Tabela wyświetlana tylko, gdy w **Strukturze** zostało wybrane urządzenie. Służy do dodawania odczytów rejestrów i konfiguracji ich parametrów - zostało to opisane w dalszej części tego rozdziału.

### Dodawanie połączenia

Połączenie określa sposób oraz parametry komunikacji modułu Base z urządzeniami magistrali MODBUS. W obrębie jednego połączenia może funkcjonować wiele urządzeń, muszą one jednak mieć takie same parametry transmisji. Konfiguracja połączenia przebiega następująco:

1. Dodaj połączenie klikając **+ Połączenie**
2. W oknie **Szczegóły**, w polu **Nazwa** wpisz nazwę połączenia (bez spacji i polskich znaków). Uzupełnienie tego pola nie jest niezbędne. Użyta nazwa musi być unikalna w całej konfiguracji. Nazwa jest używana w identyfikatorach, które służą do odczytu stanu urządzenia i sterowania nim.
3. Z listy **Typ** wybierz rodzaj połączenia. Do wyboru są trzy opcje:
  - **Serial**
  - **TCP**
  - **UDP**

#### Serial

Serial, czyli połączenie szeregowe oparte o standard RS-485. Wybierz tą opcję, kiedy integrowane urządzenia obsługują MODBUS w trybie RTU.

W polu **Format** ustaw parametry komunikacji: prędkość komunikacji, do wyboru: **9600**, **19200**, **38400** oraz **57600** bit/s oraz rodzaj ramki: **8N1**, **8N2**, **8E1** oraz **8O1**.



## TCP

MODBUS TCP to modyfikacja standardowego protokołu MODBUS RTU, w którym warstwę transportową opartą o RS-485 zastąpiono użyciem protokołu TCP IP. MODBUS TCP wykorzystuje sieć Ethernet jako warstwę fizyczną połączeń.

Tutaj dostępne są trzy parametry:

- **Nazwa** - Wpisz nazwę (bez spacji i polskich znaków), która jednoznacznie będzie identyfikowała ten interfejs. Nazwa interfejsu jest wykorzystywana w identyfikatorach.
- **Adres IP** - Adres IP urządzenia slave, z którym chcesz nawiązać połączenie.
- **Port** - Numer portu, na który wysyłane są pakiety MODBUS/TCP. Standardowo jest to port **502**, który jest oficjalnie zarezerwowany dla tego zastosowania.

## UDP

Niektóre urządzenia wykorzystują też niestandardowy wariant, w którym zamiast TCP użyto protokołu bezstanowego UDP. Typ **UDP** ma identyczny zestaw parametrów jak **TCP**.

## Dodawanie urządzenia

Urządzenie determinuje warunki komunikacji z konkretnym urządzeniem Modbus. Aby dodać nowe urządzenie, wybierz interfejs, do którego chcesz dodać urządzenie i kliknij **+ Urządzenie**. Następnie skonfiguruj jego parametry:

- **Nazwa** - Unikalna nazwa urządzenia (bez spacji i polskich znaków). Nazwa jest wykorzystywana w identyfikatorach.
- **Adres** - Adres urządzenia MODBUS slave. Adres urządzenia znajdziesz w karcie katalogowej urządzenia. W większości urządzeń, domyślny adres to 1. Dopuszczalny zakres adresów: 0-247, jednakże adres 0 to adres rozgłoszeniowy - wysłanie danych na urządzenie o tym adresie spowoduje rozgłoszenie tych danych do wszystkich urządzeń. Jest to szczególnie przydatne do sterowania grupowego urządzeń tego samego typu, np. wyłączenie wszystkich klimatyzatorów.
- **Opis** - Pomocniczy opis urządzenia ułatwiający jego identyfikację.
- **Użyj funkcji 16 (Write Multiple Registers) zamiast funkcji 6 (Write Single Register)** - Niektóre urządzenia Modbus nie posiadają implementacji funkcji 6 do zapisu rejestrów, a udostępniają jedynie funkcję 16. Zaznacz to pole, jeżeli dotyczy to podłączanego urządzenia.

## Konfiguracja rejestrów

Rejestry stanowią odniesienie do parametrów integrowanego urządzenia. Rejestry mogą być tylko do odczytu lub typu odczyt/zapis.

Dla przykładu, większość central wentylacyjnych udostępnia następujące rejestry: temperatury nawiewu, wywiewu, czerpni, wyrzutni, prędkość nawiewu, otwarcie bypassu.

**WAŻNE:** Protokół Modbus wymaga cyklicznego odpytywania rejestrów, aby pozyskiwać ich aktualne wartości. Rejestry odpytywane są sekwencyjnie - jeden po drugim. Dlatego im więcej urządzeń i rejestrów, tym mniejsza częstotliwość odświeżania wartości.

Tabela **Rejestry** wyświetlana jest po wybraniu urządzenia w **Strukturze**. Aby dodać nowy rejestr kliknij **Dodaj**. Aby usunąć rejestr(y), zaznacz wybrane pozycje w tabeli i kliknij **Usuń**

Dostępne są następujące parametry:

1. **Typ** - Określa rodzaj rejestru.
  - **holding register** - Rejestr wyjściowy, funkcja 3.
  - **input register** - Rejestr wejściowy (tylko do odczytu), funkcja 4.
  - **coil** - Rejestr odnoszący się do pojedynczego wyjścia binarnego, funkcja 1
  - **input** - Stan pojedynczego wejścia binarnego (tylko do odczytu), funkcja 2.
2. **Format** - Określa typ danych użyty do reprezentacji wartości rejestru:

## 9. Integracje

- **uint16** - 16-bitowy rejestr liczb całkowitych dodatnich.
  - **int16** - 16-bitowy rejestr liczb całkowitych.
  - **uint32** - 32-bitowy rejestr liczb całkowitych dodatnich.
  - **int32** - 32-bitowy rejestr liczb całkowitych.
  - **uint32ws** - 32-bitowy rejestr liczb całkowitych dodatnich, w którym słowa 16-bitowe są zamienione miejscami.
  - **int32ws** - 32-bitowy rejestr liczb całkowitych, w którym słowa 16-bitowe są zamienione miejscami.
  - **float** - 32-bitowy rejestr liczb zmiennoprzecinkowych
3. **Adres** - Unikalny numer rejestru w integrowanym urządzeniu. W przypadku adresów 32 bitowych są one zapisywane na dwóch sąsiednich rejestrach. Tabelę z adresami rejestrów należy odszukać w karcie katalogowej integrowanego urządzenia.
- WAŻNE! Wpisanie prawidłowego adresu rejestru jest kluczowe do poprawnego jego odczytu. Zapoznaj się z poniższymi wskazówkami, aby uniknąć najczęściej popełnianych błędów:**
- Niektórzy producenci podają adresy w formacie heksadecymalnym, należy wówczas ich wartość przeliczyć na liczbę dziesiętną.
  - Adresacja rejestrów wg standardu Modbus rozpoczyna się od 0, jednakże niektórzy producenci podają numery powiększone o 1. Wówczas należy adres pomniejszyć o 1.
  - Niekiedy numeracja rejestrów typu holding register podawana jest jako 4xxxx, np. 40108. Taki zapis odpowiada rejestrowi o adresie 107. Analogicznie jest w przypadku rejestrów typu input register, jednakże tam numeracja jest podawana w formie 3xxxx, np 30108, co odpowiada adresowi 107.
4. **Nazwa** - Unikalna nazwa rejestru. Nazwa rejestru jest wykorzystywana w identyfikatorach.
5. **Mnożnik** - Wartość przez którą pomnożona będzie aktualna wartość rejestru. Mnożnik może być również liczbą ułamkową, jednak należy pamiętać o użyciu kropki jako separatora dziesiętnego.
6. **Przesunięcie** - Wprowadzona tu wartość będzie dodana do aktualnej wartości rejestru.
7. **Dokładność** - W tym polu określana jest precyzja, z jaką będzie wyświetlana zawartość rejestru MODBUS. Jeżeli w tym polu wpiszesz 2, wówczas wartość rejestru będzie wyświetlana w stanie z dokładnością do dwóch miejsc po przecinku.
8. **Opis** - Pomocniczy opis rejestru. Będzie on wyświetlany w zakładce **Stan**.

## Usuwanie problemów

W tej sekcji opisujemy najczęściej występujące problemy i sposoby ich diagnozowania.

1. **Skonfigurowałem połączenie, urządzenie i rejestry, ale nie widzę ich odczytów w zakładce Stan.**
  - Na czas podłączania nowego urządzenia zalecamy tymczasowe odłączenie wszystkich pozostałych urządzeń od magistrali Modbus. (tylko Modbus RTU).
  - Sprawdź miganie diód komunikacyjnych na module Serial-4MB. W przypadku prawidłowej komunikacji powinny migać diody ACT oraz naprzemiennie TX oraz RX. Jeżeli dioda RX nie miga, oznacza to brak komunikacji z integrowanym urządzeniem.
  - Sprawdź parametry komunikacji - upewnij się, że ustawienia prędkości i formatu ramki są takie same w module Base i podłączanym urządzeniu (tylko dla Modbus RTU).
  - Upewnij się, że adres urządzenia został ustawiony i czy nie pokrywa się z innym urządzeniem na magistrali.
  - Upewnij się, że wprowadzone są nazwy dla połączenia, urządzenia i rejestrów oraz że nie zawierają polskich znaków i spacji.
  - Upewnij się, że rejestry mają prawidłową numerację - zalecamy skonfigurowanie pojedynczego rejestru, którego wartość jest łatwa do weryfikacji.

- Spróbuj zamienić przewody magistrali (tylko Modbus RTU). W niektórych przypadkach producenci urządzeń na schematach prezentują połączenie przewodów A-A oraz B-B, a w niektórych przypadkach wymagane jest połączenie na krzyż.
- Upewnij się, że zaciski COM modułu Serial-4MB oraz integrowanego urządzenia są podłączone do magistrali. Jest to szczególnie istotne w przypadku dłuższych magistral, które narażone są na dużo większe zakłócenia transmisji. (tylko Modbus RTU).
- Sprawdź, czy dokumentacja podłączanego urządzenia wspomina o konieczności użycia rezystora terminującego. W module Serial-4MB służą do tego kołki `TERM` obok terminala śrubkowego.

# 9.3. DALI

DOMIQ/Base w połączeniu z modułem DOMIQ/Light umożliwia sterowanie instalacjami oświetleniowymi opartymi o protokół DALI. Protokół został zaprojektowany specjalnie z myślą o sterowaniu oświetleniem. Jego najważniejsze cechy to:

- możliwość sterowania jasnością oświetlenia oraz odczytywania aktualnej jasności;
- niezawodność i prostota okablowania;
- w pełni zautomatyzowany proces wykrywania i adresowania opraw oświetleniowych;
- mechanizm wykrywania usterek w działaniu oświetlenia;
- możliwość zapisywania najczęściej używanych nastaw w formie scen (max. 16 scen);
- możliwość grupowania opraw dla łatwiejszego zarządzania (max. 16 grup);
- możliwość zaprogramowania jasności, którą ma wywołać lampa po przywróceniu zasilania;
- możliwość zaprogramowania jasności, którą ma wywołać lampa po wystąpieniu błędu/awarii;

Ponadto w DOMIQ dodaliśmy:

- Obsługę komendy "przełącz", która naprzemiennie włącza/wyłącza balast.
- Obsługę umożliwiającą sterowanie jasnością w oparciu o długie naciśnięcia przycisków. Sposób działania jest następujący: długie naciśnięcie rozpoczyna rozjaśnianie, puszczenie przycisku zatrzymuje proces, kolejne długie naciśnięcie rozpoczyna ściemnianie.
- Możliwość jednoczesnego wysyłania komend do wszystkich balastów.

Implementacja protokołu DALI w systemie DOMIQ pozwala stworzyć system oświetleniowy, który działa niezawodnie i bez opóźnień w reakcji na komendy.

Komendy sterujące balastami DALI zostały opisane w rozdziale 12. Identyfikatory.

## Etapy integracji

### 1. Wykonanie okablowania.

Magistrala DALI powinna mieć topologię gwiazdy lub liniową. Do jej wykonania wystarczy przewód dwużyłowy. Maksymalna dopuszczalna długość magistrali DALI to 300m. Należy również ściśle przestrzegać wytycznych odnośnie przekrojów używanego okablowania. W przypadku przekroczenia maksymalnej długości i/lub użycia przewodów o niewłaściwym przekroju, instalacja może działać nieprawidłowo (np. niewykrywanie balastów lub niereagowanie na komendy) lub nie działać wcale. Z uwagi na ewentualność późniejszej rozbudowy instalacji zalecamy stosowanie przewodów o przekroju 1.5mm<sup>2</sup>.

Długość magistrali	Minimalny zalecany przekrój przewodów
do 100m	0,5mm <sup>2</sup>
100-150	0,75mm <sup>2</sup>
150-300	1,5mm <sup>2</sup>

### 2. Podłączenie balastów.

### 3. Adresowanie balastów.

Nowe balasty są bez adresów, dlatego niezbędne jest ich adresowanie. W systemie DOMIQ możliwe jest uruchomienie automatycznej procedury wykrywania i nadawania adresów podłączonym balastom, a także zmiana adresów poszczególnych w późniejszym czasie. Ponadto dostępny jest mechanizm adresowania balastów przyłączonych do instalacji później, które jeszcze nie mają adresów.

### 4. Konfiguracja balastów.

Na tym etapie konfigurowane są parametry poszczególnych opraw, takie jak: maksymalna i minimalna jasność, przynależność do grup, zapisanie scen świetlnych, itd.

## Interfejs konfiguracyjny

Interfejs konfiguracyjny znajdujący się w zakładce **DALI** podzielony jest na dwie kolumny. W lewej kolumnie wyświetlana jest tabela balastów podłączonych do magistrali, natomiast prawa strona służy do konfiguracji i sterowania.

### 1. Tabela balastów

W tabeli wyświetlane są dane balastów, które są podłączone do magistrali i mają nadany prawidłowo adres.

**WAŻNE!** Tabela nie jest odświeżana automatycznie. Aby odświeżyć dane, kliknij przycisk **Odśwież**.

- **Evg** - W tej kolumnie wyświetlany jest adres balastu (0 do 63)
- **Jasność** - Bieżąca jasność balastu (0 do 254).
- **Włączony** - Flaga statusowa informująca o załączeniu balastu. 0 – wyłączony, 1 – włączony.
- **Brak zasilania** – Flaga statusowa informująca o awarii zasilania balastu. 0 – OK, 1 – awaria.
- **Awaria** – Flaga statusowa informująca o awarii balastu. 0 – OK, 1 – awaria.

Przycisk **Skanuj** służy do uruchomienia procedury wykrywania balastów mających nadane adresy, ale podłączonych do magistrali w późniejszym czasie.

Wybranie konkretnego balastu w tabeli ładuje jego dane w oknie **Pojedynczy balast** w prawej części interfejsu.

### 2. Wszystkie balasty

Panel umożliwia sterowanie i adresowanie całej instalacji.

#### Sterowanie

Sekcja **Sterowanie** zawiera przyciski umożliwiające sterowanie całą instalacją oświetleniową DALI. Komendy są wysyłane do wszystkich balastów podłączonych do magistrali (również do tych bez przypisanych adresów). Dostępne są następujące opcje:

- Pole edycyjne i przycisk **Ustaw** - Umożliwia wysterowanie wszystkich balastów na określoną jasność (0-254).
- **Włącz** – Włącza wszystkie lampy. Lampy zostaną wysterowane do wartości ustawionego w poszczególnych balastach poziomu maksymalnego.
- **Wyłącz** – Wyłącza wszystkie lampy.
- **Maksimum** – Ustawia jasność na maksimum.
- **Minimum** – Ustawia jasność na minimum
- **+** – Skokowe rozjaśnianie. Zmiany są zależne od ustawień parametru **Skok jasności** poszczególnych balastów. Komenda nie załącza wyłączzonego balastu.
- **-** – Skokowe ściemnianie. Zmiany są zależne od ustawień parametru **Skok jasności** poszczególnych balastów. Wysłanie komendy, gdy jasność jest ustawiona na minimum nie powoduje wyłączenia balastu.

#### Adresowanie

- **Wszystkie** - Uruchamia procedurę nadawania adresów wszystkim balastom w instalacji (również tym bez nadanych adresów). Po jej uruchomieniu wszystkie balasty w instalacji (z adresami i bez) otrzymają nowe, losowe adresy. Uruchomienie tej procedury w istniejącej instalacji spowoduje skasowanie dotychczasowej adresacji.
- **Nieprzypisane** - Uruchamia procedurę automatycznego adresowania tylko tych balastów, które nie mają przyporządkowanego adresu.
- **Adres początkowy** - Pozwala ustalić początkowy adres względem, którego zainicjowana zostanie procedura adresowania. Ma to szczególne znaczenie w przypadku, gdy potrzebne jest ponowne zaadresowanie tylko części instalacji.

## 9. Integracje

**WAŻNE!** Pomyślne przypisanie adresu zostanie zasygnalizowane zapaleniem danej lampy na maksymalną jasność.

### 3. Pojedynczy balast

#### Sterowanie

Zawiera zestaw przycisków identyczny, jak w przypadku sterowania całą instalacją. Komendy są wysyłane do wybranego balastu.

#### Adresowanie

- Pole edycyjne i przycisk **Ustaw adres** - umożliwia zmianę bieżącego adresu balastu.
- **Usuń adres** - Usuwa bieżący adres balastu. Aby nadać ponownie adres, należy uruchomić procedurę adresowania naciskając przycisk **Nieprzypisane**.

#### Parametry

Ta część interfejsu służy do konfiguracji parametrów pracy wybranego balastu DALI. Aby wprowadzone nastawy zostały zapisane w pamięci balastu, balast musi być wyłączony.

1. **Skok** – Określa skokową zmianę jasności przy wywołaniu komend `up` i `down`. Dopuszczalna wartość od 1 do 15. Poniższa tabela zawiera wartości skoku i przyporządkowaną im liczbę wywołań (np. naciśnięcie przycisku) niezbędną do pełnej zmiany jasności. W zależności od producenta i modelu balastu liczba ta może się różnić z uwagi na wartość fizycznej minimalnej jasności.

Wartość skoku	Liczba wywołań
1	4
2	6
3	8
4	10
5	15
6	20
7	29
8	43
9	51
10	85
11	102
12	127
13	170
14	254
15	508

2. **Rampa** - Czas zmiany jasności z wartości bieżącej na zadaną. Rampa jest niezależna od poziomu zmian, dlatego przy sterowaniu wielu lamp, które mają ustawioną tę samą rampę, zmiana zakończy się w tym samym momencie. Poniższa tabela przedstawia zakres dopuszczalnych wartości i czasy im przyporządkowane.

Wartość rampy	Czas przejścia [s]
0	0,5
1	0,707
2	1
3	1,414

4	2
5	2,828
6	4
7	5,657
8	8
9	11,314
10	16
11	22,627
12	32
13	45,255
14	64
15	90,510

- Maksimum** – Maksymalny dopuszczalny poziom jasności. Jeśli nowe maksimum jest mniejsze od poziom minimalnego, wówczas poziom minimalny zostanie ustawiony jako nowe maksimum. Dopuszczalna wartość od 0 do 254. Wartość domyślna: 254.
- Minimum** – Minimalny dopuszczalny poziom jasności. Minimalny poziom jasności nie może być mniejszy niż minimalny fizyczny poziom jasności (zależny od modelu i producenta balastu). W przypadku próby ustawienia wartości poniżej minimalnego poziomu fizycznego, wartość zostanie automatycznie dopasowana do poziomu fizycznego minimum. Jeżeli wartość minimalna zostanie ustawiona powyżej poziomu maksymalnego, wówczas dotychczasowy poziom maksymalny zostanie ustawione jako nowe minimum. Dopuszczalna wartość od 0 do 254. Domyślnie poziom minimum jest równy minimalnemu fizycznemu poziomowi danego balastu.
- Poziom po załączeniu** – Poziom jasności, na który lampa zostanie wysterowana po podaniu zasilania. Dopuszczalna wartość od 0 do 255. W przypadku wprowadzenia wartości 255, w zależności od producenta balastu, zostanie przywrócona ostatnio ustawiona jasność lub wartość ostatniej wysłanej komendy zmieniającej jasność. Domyślna wartość: 254.
- Poziom przy błędzie** – Poziom jasności, na który lampa zostanie wysterowana w przypadku awarii. Dopuszczalna wartość od 0 do 255. Jeżeli ustawiona zostanie wartość 255, wówczas przy awarii jasność nie ulegnie zmianie. Wartość domyślna: 254.

## Sceny i grupy

Zgodnie ze standardem DALI, każdy balast może mieć zapisanych 16 scen świetlnych i być przypisany do maksymalnie 16 grup. Tabela zarządzania scenami i grupami podzielona jest na pięć kolumn:

- Scena** – Zawiera numery poszczególnych scen (od 0 do 15).
- Jasność** – W tej kolumnie możesz wprowadzić nową jasność, którą chcesz zapisać do sceny. Kolumna wyświetla również bieżące wartości poszczególnych scen.
- Zapisz scenę** – Zawiera przyciski **Zapisz**, które zapisują do wybranej sceny wartość wpisaną w komórce **Jasność**.
- Załaduj scenę** – Zawiera przyciski **Załaduj** do wywoływania poszczególnych scen.
- Grupy** – W tej kolumnie możesz wybrać grupy, do których balast ma być przypisany.

## Sterowanie z LCN

Moduł Base umożliwia przekazywanie komend wprost z instalacji LCN do modułu DOMIQ/Light i tym samym sterowanie instalacją DALI do niego podłączoną. Dzięki temu możliwe staje się sterowanie np. wprost z przycisków ściennych LCN, jako reakcja na naruszenie czujki PIR lub w oparciu o czujnik nasłonecznienia, itd.

Konfiguracja ogranicza się wyłącznie do odpowiedniego sparametryzowania modułu LCN w oprogramowaniu LCN-Pro.

## 9. Integracje



Jako przykład przedstawimy, jak włączyć lampę przy użyciu przycisku ściennego przypisanego do klawisza A1. Akcja „**krótko**” będzie załączać balast o adresie 0.

1. Wybierz moduł LCN, do którego podłączone są przyciski.
2. Jako adresata akcji **A1 krótko** wybierz moduł Base (domyślnie adres 254).
3. Jako typ komendy wybierz: **DALI**.
4. Kolejnym krokiem jest wybranie adresu balastu. **UWAGA!** W systemie LCN numeracja balastów zaczyna się od 1, natomiast w DOMIQ jest liczona od 0. Zatem aby wystawiać lampę o adresie 0, wybieramy 1 w LCN-Pro.
5. Ostatnim krokiem jest wybranie komendy DALI, która zostanie wysłana do modułu DOMIQ/Light. W naszym przypadku wybieramy **Ustaw** i wpisujemy wartość 100%.



## 9.4. Wykresy

Możliwość zobrazowania danych na wykresie jest jedną z bardziej pożądanых cech inteligentnego budynku. Moduł Base umożliwia integrację z oprogramowaniem Prometheus.io, i Grafana, które w dużej mierze stanowią doskonałe narzędzie do akwizycji i wizualizacji danych. W ten sposób możliwe jest zbieranie i wizualizowanie danych dotyczących dowolnej zmiennej występującej w systemie DOMIQ. Z łatwością zwizualizujesz np. zużycie/produkcję energii elektrycznej, wody lub gazu obejrzyj przebiegi temperatury lub parametrów pracy systemu rekuperacji w dowolnym zakresie czasowym.

Zakładka **Wykresy** służy do definiowania tzw. metryk, które w dalszym etapie służą do pobierania z modułu Base danych do prezentacji na wykresach. Metryki to nic innego, jak definicje zmiennych, które Grafana będzie odczytywała z modułu Base.

Zakładka ma trzy kolumny: **Identyfikator**, **Nazwa** oraz **Opis**. Aby dodać nową metrykę kliknij przycisk **Dodaj** i skonfiguruj ją następująco:

- **Identyfikator** - wpisz nazwę identyfikatora, który jest dostępny w systemie DOMIQ.
- **Nazwa** - Uzupełnienie tej kolumny jest opcjonalne, ale przez nas bardzo zalecane. W tej kolumnie możesz wpisać nazwę, która będzie opisywała czym jest dany identyfikator.

**Przykład:** Przyjmijmy, że użyto identyfikatora `LCN.value.0.10.r1`. Ta nazwa za wiele nie mówi, o tym, co kryje się w zmiennej R1 modułu LCN z adresem 10. Natomiast jeżeli w kolumnie **Nazwa** wpiszesz `temperatura_salon`, wówczas dużo łatwiej będzie zrozumieć przeznaczenie danej metryki. Jest to sporym ułatwieniem zwłaszcza przy definiowaniu wykresów w oprogramowaniu Grafana.

Jeżeli kolumna **Nazwa** nie zostanie uzupełniona dla danej metryki, wówczas w czasie tworzenia wykresu w oprogramowaniu Grafana, należy odwoływać się wprost do nazwy identyfikatora, ale w miejsce kropek należy używać znaków podkreślenia i małych liter. Nie wolno tu używać spacji i polskich znaków!

**Przykład:** Jeżeli użyjesz identyfikatora `LCN.value.0.10.r1`, wówczas w definicji wykresu należy użyć `lcn_value_0_10_r1`. Oprogramowanie Grafana ułatwia nam pracę, ponieważ ma mechanizm autouzupełniania nazw metryk.

- **Opis** - Tu możesz wpisać dowolny opis dotyczący metryki.

Wg powyższej procedury zdefiniuj tyle metryk ile potrzebujesz, a na końcu zapisz zakładkę **Wykresy**. Aby usunąć metryki, zaznacz je w tabeli i kliknij przycisk **Usuń**.

Przycisk **Importuj** umożliwia zaimportowanie zestawu metryk z pliku .xml zapisanego na dysku.

Analogicznie przycisk **Eksportuj** umożliwia zapisanie na dysku metryk utworzonych w zakładce **Wykresy**.

Więcej o wykresach i integracji z oprogramowaniem Prometheus.io i Grafana przeczytasz w samouczku: <https://www.domiq.pl/wiki/wykresy-w-domiq-integracja-z-grafana/>

# 9.5. DMX

Zakładka **DMX** służy do definiowania scen świetlnych dla oświetlenia używającego standardu DMX-512. W obrębie sceny możesz dodać dowolną liczbę kanałów DMX wraz z jasnościami dla poszczególnych kanałów. W ten sposób z łatwością możesz zaprogramować nawet bardzo złożone sceny świetlne.

Aby zdefiniować scenę wykonaj następujące kroki:

1. Kliknij **Dodaj** w panelu **Sceny**.
2. W polu **Nazwa** wpisz unikalną nazwę sceny. Nazwa nie może mieć spacji i polskich znaków. W polu **Opis** możesz wpisać opis ułatwiający identyfikację danej sceny.
3. W panelu **Kanały przypisane do sceny** należy zdefiniować kanały DMX, które mają być powiązane z daną sceną.
  - **Numer** - Tu wpisz numer kanału DMX (1-256).
  - **Wartość** - Tu wpisz docelową jasność (0-100).
  - **Rampa** (opcjonalnie) - W tym polu możesz wpisać rampę (czas przejścia) dla zmiany jasności (0-128).

W panelu **Opis kanałów** możesz dodać opisy dla poszczególnych kanałów DMX, aby ułatwić ich późniejszą identyfikację. Uzupelnienie tabeli nie ma związku z tworzeniem scen.

Przycisk **Importuj** umożliwia zaimportowanie scen z pliku `.xml`. Analogicznie przycisk **Eksportuj** umożliwia zapisanie do pliku bieżącej konfiguracji.

Komendy sterujące scenami zostały opisane w rozdziale 12. Identyfikatory.

## 9.6. BACnet

Moduł Base umożliwia integrację z systemem BACnet. Funkcje integracyjne zostały opracowane zgodnie z wytycznymi normy ASHRAE-135-2012 i załącznikiem J „BACnet/IP” tej normy. Z punktu widzenia instalacji BACnet, moduł Base pełni rolę B-ASC (BACnet Application-Specific Controller) – sterownika specjalnego zastosowania. Sterowniki specjalnego zastosowania są tworzone do realizowania z góry określonej funkcjonalności, zaprogramowanej przez producenta urządzenia. W takich urządzeniach, konfiguracja po stronie użytkownika jest ograniczona wyłącznie do zmiany parametrów. W przypadku modułu Base, zaimplementowana została funkcjonalność w dwóch obszarach przewidzianych w normie:

1. **Data Sharing** - Wymiana danych z innymi urządzeniami BACnet obecnymi w instalacji.
  - **Data Sharing-ReadProperty-B (DS-RP-B)** - Odczyt wartości (np. temperatura, stan przełącznika).
  - **Data Sharing-ReadPropertyMultiple-B (DS-RPM-B)** - Odczyt wielu wartości jednocześnie.
  - **Data Sharing-WriteProperty-B (DS-WP-B)** - Zapis wartości.
  - **Data Sharing-COV-B (DS-COV-B)** - Raportowanie o zmianie wybranej wartości na żądanie innego urządzenia BACnet.
2. **Device and Network Management**
  - **Device Management-Dynamic Device Binding-B (DM-DDB-B)** - Dynamiczne przypisywanie i zarządzanie urządzeniami.

W aspekcie integracji podsystemów inteligentnego budynku, Base pełni rolę bramki sprzęgającej system BACnet z dowolnym podsystemem (LCN, Satel, Modbus itd.). Możliwość integracji z systemem BACnet dostępna jest po wykupieniu dodatkowej licencji parującej. Klucz licencyjny należy dodać w zakładce **Ustawienia**.

Dostępne funkcje integracyjne umożliwiają:

- **Zapis wejść analogowych, binarnych i wielostanowych.** W tym przypadku dowolny identyfikator w stanie modułu Base może być źródłem wartości dla wejścia w systemie BACnet, np. wartość temperatury zmierzonej przez czujnik LCN, stan kontaktronu, przełącznika itd.
- **Dwukierunkowy zapis wyjść analogowych, binarnych i wielostanowych.** Dzięki temu możliwe jest dwukierunkowe przekazywanie wartości pomiędzy urządzeniem BACnet a dowolnym innym urządzeniem w instalacji inteligentnego budynku. Dla przykładu, jeżeli wyjście w systemie BACnet powiążemy z wyjściem ściemnianym w systemie LCN, wówczas w przypadku zmiany wartości wyjścia w systemie BACnet, wyjście LCN również zmieni swoją wartość. Działa to również odwrotnie, zatem zmiana wartości wyjścia w module LCN spowoduje zmianę w systemie BACnet.
- **Dwukierunkowy zapis wartości binarnych i analogowych.**

W dalszej części tego podrozdziału opisany został interfejs konfiguracyjny zakładki **BACnet**.

### Podgląd zmiennych BACnet

W tabeli widoczne są wszystkie zdefiniowane zmienne wraz z ich parametrami. Biorąc pod uwagę funkcjonalność, jaką realizuje moduł Base, zmienne umożliwiają odczyt/zapis dowolnego identyfikatora, będącego w stanie Base, innym urządzeniom na magistrali BACnet.

Aby dodać nową zmienną, kliknij przycisk **Dodaj** i wybierz typ zmiennej.

Dostępne typy:

- **Wejście binarne** - Reprezentuje stan fizycznego wejścia binarnego w instalacji. Może być w jednym z dwóch stanów: *aktywny* lub *nieaktywny*.  
Przykładowe zastosowanie: czujka ruchu, kontaktron, przycisk.
- **Wyjście binarne** - Reprezentuje stan fizycznego wyjścia binarnego w instalacji. Może być w jednym z dwóch stanów: *aktywny* lub *nieaktywny*.  
Przykładowe zastosowanie: wentylator, pompa, oświetlenie.
- **Wartość binarna** - Reprezentuje stan wartości binarnej, które przechowywana jest w pamięci urządzenia BACnet. Może być w jednym z dwóch stanów: *aktywny* lub *nieaktywny*.

## 9. Integracje

- **Wejście analogowe** - Reprezentuje stan fizycznego wejścia analogowego w instalacji. Przykładowe zastosowanie: odczyt temperatury, wilgotności itd.
- **Wyjście analogowe** - Reprezentuje stan fizycznego wyjścia analogowego w instalacji. Przykładowe zastosowanie: nastawy temperatury, sterowanie oświetleniem itp.
- **Wartość analogowa** - Reprezentuje stan wartości analogowej w instalacji, która jest przechowywana w pamięci urządzenia BACnet.
- **Wejście wielostanowe** - W swoim rodzaju podobne do wejść binarnych, z tą różnicą, że mogą przyjmować wiele stanów. Wartość wejścia wielostanowego najczęściej odnosi się do wartości wirtualnej, rzadziej do fizycznego wejścia w urządzeniu. Przykładowe zastosowanie: zobrazowanie aktualnego biegu centrali wentylacyjnej, poziom w zbiorniku.
- **Wyjście wielostanowe** - Zmienna, która może przyjmować jeden z określonych stanów. Najczęściej jest to zmienna przechowywana w pamięci urządzenia, rzadziej fizyczne wyjście w urządzeniu. Przykładowe zastosowanie: wybór trybu pracy klimatyzacji (grzanie/chłodzenie/nawiew).

## Urządzenie

W tym oknie należy wprowadzić parametry konfigurujące moduł Base. do pracy w systemie BACnet.

- **Nazwa** – Nazwa pod którą moduł Base będzie widoczny w systemie BACnet.
- **Adres** – Unikalny adres urządzenia w systemie BACnet, który zostanie przyporządkowany modułowi Base. W systemie BACnet może pracować maksymalnie 4194304 urządzenia, co oznacza, że każde może mieć nadany adres w zakresie od 0 do 4194303. Przyjęło się stosowanie adresacji zgodnie z konwencją XXFFBBB, gdzie:
  - XX to numer urządzenia w zakresie od 0 do 40.
  - FF to numer piętra w zakresie od 01 do 35. Wartość 00 jest zarezerwowana dla sieci szkieletowej budynku.
  - BBB oznacza numer budynku w instalacji w zakresie od 0 do 654.

Przykładowo urządzenie o adresie 1234567 oznacza urządzenie nr 12 na piętrze 34. w budynku nr 567. Ten system numeracji ma dużą zaletę - pozwala łatwo i szybko ustalić fizyczną lokalizację urządzenia. Tak przyjęta numeracja dopuszcza 41 urządzeń na każdym piętrze, 35 pięter w budynku i aż 655 budynków. W przypadkach wymagających innej numeracji, można zastosować inną konwencję adresowania urządzeń.

- **Opis** (opcjonalne) – Krótki opis urządzenia, który będzie widoczny w narzędziach konfiguracyjnych systemu BACnet.
- **Lokalizacja** (opcjonalne) – Informacja o lokalizacji urządzenia. Przydatne w przypadku bardziej rozległych instalacji BACnet.

## Ustawienia zmiennej

Ten panel wyświetlany jest po wybraniu dowolnej zmiennej w tabeli zmiennych. Wyświetlane w nim ustawienia są zmienne i zależą od typu wybranej zmiennej.

Dla wszystkich typów zmiennych wspólne są następujące parametry:

- **ID** – Unikalny identyfikator zmiennej w obrębie urządzenia.
- **Nazwa** – Nazwa zmiennej. Może być to dowolna nazwa (bez polskich znaków). Nazwa będzie widoczna dla innych urządzeń w systemie BACnet.
- **Kanał** – Identyfikator w systemie DOMIQ, który ma być powiązany z tą zmienną. Bieżąca stan identyfikatora stanowi źródło wartości przekazywanych do zmiennej w systemie BACnet. W przypadku zmiennych będących wejściami, wartość identyfikatora jest tożsama z wartością wejścia w systemie BACnet. Natomiast w przypadku zmiennych będących wartościami lub wyjściami, przekazywanie danych odbywa się dwukierunkowo.
- **Opis** - Opis ułatwiający identyfikację zmiennej.

Parametry zależne od typu zmiennej:

1. Zmienne analogowe:

- **Jednostka** – Jednostka zmiennej w systemie BACnet.
- **Mnożnik** – W tym polu można wpisać dowolną liczbę dodatnią, również ułamkowa, przez którą zostanie pomnożona wartość zmiennej. W przypadku mnożników ułamkowych należy używać kropki jako separatora dziesiętnego.
- **Przesunięcie** – W komórce **Przesunięcie** można wprowadzić dowolną liczbę, także ułamkową i ujemną, która zostanie dodana do zmiennej.

## Stany (tylko wejścia i wyjścia wielostanowe)

Wejścia i wyjścia wielostanowe mogą przyjmować jeden z zadeklarowanych stanów. Panel **Stany** służy do definiowania stanów dla wejść i wyjść wielostanowych. Jeżeli przypisany do zmiennej identyfikator w systemie DOMIQ osiągnie wartość odpowiadającą jednemu ze stanów, wówczas w systemie BACnet zmienna będzie sygnalizowała stan przypisany do tej wartości.

Aby dodać nowy stan, kliknij **Dodaj** i uzupełnij jego właściwości.

- **Nazwa** - Nazwa stanu.
- **Wartość** - Wartość, która jest przypisana do danego stanu.
- **Komenda** (tylko wyjście wielostanowe) - Dodatkowa komenda, która zostanie wykonana, gdy ustalony zostanie określony stan.

## Subskrypcje

Protokół BACnet zaprojektowany został do realizacji nawet bardzo rozległych instalacji automatyki. Każde z urządzeń może mieć zdefiniowanych setki, a nawet tysiące zmiennych! Moduł Base pełni rolę *Data Sharing* (wymiana danych). Urządzenia tego typu pełnią rolę serwera udostępniając dane do odczytu i zapisu innym urządzeniom w instalacji BACnet. Inne urządzenia (typu klient) chcąc pozyskać z nich dane, muszą o te dane odpytać (ang. data polling). W przypadku, gdy urządzenie ma skonfigurowanych wiele zmiennych, pociąga to za sobą konieczność cyklicznego odpytywania każdej zmiennej z osobna, niezależnie od tego, czy zaszła w niej zmiana wartości, czy nie. Jest to bardzo kosztowne pod względem czasu i zasobów serwera (głównie CPU).

Projektanci protokołu BACnet, aby zaradzić temu problemowi utworzyli koncepcję subskrypcji i notyfikacji o zmianach. Urządzenie, które jest zainteresowane wybranymi zmiennymi tworzy subskrypcję na wskazane identyfikatory zmiennych, a urządzenie docelowe rozsyła notyfikacje o zmianach wartości zmiennych do zarejestrowanych subskrybentów. Taki model wymiany danych w znaczący sposób ogranicza ruch sieciowy i eliminuje konieczność cyklicznego odpytywania o zmienne.

Moduł Base posiada implementację tego mechanizmu. Do tworzenia subskrypcji służy tabela **Subskrypcje**. Każda zmienna BACnet zdefiniowana w module Base może mieć własny zestaw subskrypcji - czyli informacji o adresach urządzeń BACnet, do których ma być wysłana notyfikacja o zmianie wartości danej zmiennej.

Aby dodać nową subskrypcję kliknij przycisk **Dodaj** i skonfiguruj parametry:

- **Adres** - Adres urządzenia BACnet, do którego ma zostać wysłana notyfikacja z nową wartością zmiennej.
- **Potwierdzenie** - Jeżeli to pole jest zaznaczone, wówczas subskrybent zobowiązany jest potwierdzić odebranie notyfikacji. Generuje to dodatkowy ruch sieciowy i obciążenie dla obu urządzeń. Domyślnie ta opcja jest wyłączona. Zaznacz ją, jeżeli notyfikacja dotyczy szczególnie ważnych danych i wymaga ona dodatkowych środków bezpieczeństwa.



### 9.7. u::Lux

Mnogość sterowanych urządzeń, sceny świetlne, sterowanie ogrzewaniem, wentylacją, klimatyzacją, to tylko część możliwości inteligentnego domu. Zastosowanie w tym celu tradycyjnych włączników bardzo komplikuje realizację części funkcjonalności lub zupełnie ją uniemożliwia. Wychodząc naprzeciw tym wymaganiom oferujemy włączniki u::Lux. W niewielkim rozmiarze tradycyjnego włącznika zamknięta została wielka funkcjonalność, dzięki której zrealizujesz nawet bardzo złożone funkcje, a także z łatwością dopasujesz je do swoich potrzeb. Jeżeli w przyszłości Twoje preferencje się zmieniają, wówczas bez problemu zmienisz konfigurację włącznika u::Lux, aby stale spełniała Twoje oczekiwania.

Włączniki u::Lux wyposażone są w cztery klawisze mechaniczne oraz kolorowy wyświetlacz. Każdy włącznik u::Lux może mieć zaprogramowanych do 128 klawiszy wirtualnych (32 strony po 4 klawisze). Natomiast każdy klawisz może sterować dowolnym urządzeniem/zmienną dostępną z poziomu modułu DOMIQ/Base. Klawisze wyposażone są w diody RGB LED, które mogą sygnalizować stan klawisza przez świecenie lub miganie. Klawisze rozróżniają krótkie i długie naciśnięcie, a także moment puszczenia klawisza. Dzięki temu w łatwy sposób można zrealizować np. funkcję ściemniania oświetlenia w sposób znany z aplikacji DOMIQ/Remote lub z włączników ściennych LCN. Na ekranie wyświetlane są ikony oraz informacje o sterowanych urządzeniach/zmiennych. Treści wyświetlane na ekranie są w pełni programowalne, dlatego w każdej chwili możesz zmienić wygląd i sposób działania włącznika, aby spełniał Twoje oczekiwania. Do dyspozycji jest domyślny zestaw ikon, jednakże istnieje możliwość użycia własnych grafik, aby jeszcze lepiej dopasować włączniki do Twoich potrzeb i upodobań. Włączniki u::Lux pasują do ramek tradycyjnych włączników oświetlenia, dzięki czemu z większą łatwością dopasujesz ich wygląd do wnętrza budynku.



Włącznik u::Lux wyposażony jest w czujnik jasności oświetlenia, który może posłużyć do realizacji algorytmu sterowania oświetleniem w oparciu o jasność w pomieszczeniu.

W celu oszczędzania energii, ekran włącznika może zostać wygaszony. Jego ponowne załączenie nastąpi automatycznie, gdy włącznik wykryje zbliżającego się użytkownika (włącznik posiada wbudowany czujnik zbliżeniowy).

Włącznik wyposażony jest w głośnik, dzięki czemu użytkownicy inteligentnego budynku mogą być informowani o zdarzeniach w systemie za pomocą dźwięków i komunikatów głosowych np. ostrzeżenie o pozostawionych otwartych oknach w momencie wychodzenia z domu. Włącznik u::Lux obsługuje pliki dźwiękowe w formacie MP3 oraz WAV (tylko mono, 16-bit, próbkowanie 22050Hz, nieskompresowany PCM).

Ponadto do włącznika u::Lux można podłączyć moduły rozszerzeń:

- czujnik temperatury,
- czujnik temperatury i wilgotności,

- czujnik temperatury, wilgotności i dwutlenku węgla,
- czujnik temperatury, wilgotności i detektor ruchu (wykrywa ruch w promieniu około 5m)
- moduł dwóch wejść cyfrowych.

## Komunikacja

Włączniki u::Lux komunikują się poprzez protokół UDP przy użyciu sieci LAN. Sieć LAN służy także do ich zasilania, zatem wymagany jest switch PoE lub dedykowany zasilacz marki u::Lux.

Włączniki po podłączeniu do sieci LAN i otrzymaniu adresu IP są automatycznie wykrywane przez moduł DOMIQ/Base. Ewentualna późniejsza zmiana adresu IP zostanie także automatycznie wykryta przez moduł Base. Każdy włącznik u::Lux ma swój unikalny identyfikator w obrębie instalacji, co bardzo upraszcza rozróżnianie naciśnięć klawiszy, a także sterowanie funkcjami poszczególnych włączników.

## Konfiguracja

Do konfiguracji włączników służy dedykowane oprogramowanie u::Lux Config. Z wykorzystaniem u::Lux Config m.in utworzysz strukturę klawiszy (strony) i zaprogramujesz sposób ich działania, wybierzesz ikony, przypiszesz adresy IP włącznikom itd. Opis konfiguracji włączników przy użyciu u::Lux Config nie jest częścią tego dokumentu. Wideo samouczki przedstawiające oprogramowanie u::Lux Config znajdziesz tutaj: [Samouczki u::Lux Config](#)

Podstawowym elementem sterującym we włącznikach u::Lux jest aktor. Przez pojęcie aktora należy rozumieć zmienną o numerze unikalnym w całej instalacji, do której odnosić się może dowolna liczba klawiszy w dowolnej liczbie włączników – numer aktora jest globalny w całym projekcie (instalacji). O aktorze można też myśleć jako o unikalnym identyfikatorze urządzenia lub zmiennej, którą dany klawisz lub grupa klawiszy ma sterować. W obrębie projektu może być 65535 aktorów (od 1 do 65535). Pojedynczy włącznik u::Lux może wykorzystać do 128 aktorów.

Integracja pomiędzy DOMIQ/Base a włącznikami u::Lux polega na tym, że moduł Base potrafi odbierać komendy wysyłane do aktorów, a także sam zmieniać ich stan. W tym celu należy utworzyć mapowanie, gdzie numer aktora jest przypisany do danego identyfikatora w obrębie modułu Base. Z tak utworzonymi powiązaniem możliwe jest sterowanie urządzeniami i zmiennymi dostępnymi w module Base z poziomu włączników u::Lux. Dzięki globalności aktorów, raz utworzone powiązanie pozwala na sterowanie z dowolnej liczby włączników, które mają odwołanie do danego aktora. Mapowanie jest niezbędne, aby włączniki u::Lux prezentowały bieżący stan urządzeń i zmiennych dostępnych w module Base, którymi sterują.

W dalszej części tego podrozdziału omówiony zostanie interfejs konfiguracyjny, który umożliwia tworzenie mapowania w module DOMIQ/Base.

Zakładka **ULUX** podzielona jest na dwie części. Lewa część interfejsu to **Lista aktorów**. Zawiera ona wszystkie numery aktorów, dla których utworzone zostało mapowanie. Nad tabelą są dwa przyciski: **Dodaj**, który umożliwia dodanie nowego aktora do listy oraz **Usuń**, który służy do usuwania aktorów z listy. Prawa strona interfejsu zawiera tabelę **Szczegóły**, która służy do konfigurowania parametrów mapowania dla danego aktora. Dostępne są następujące pola:

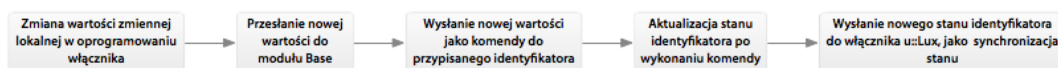
- **ID aktora (pole wymagane)** – Unikalny numer aktora, który został nadany w oprogramowaniu uLux::Config.
- **Kanał (pole wymagane)** – Identyfikator w obrębie modułu Base, z którym dany aktor ma być powiązany. Identyfikator s
- **Opis** – Pomocniczy opis, do łatwiejszej identyfikacji aktora.
- Pole wyboru **Wartość rzeczywista/Wartość edytowalna (pole wymagane)**.

Aktory mogą pracować w dwóch trybach, jako wartość edytowalna lub jako wartość rzeczywista. Bardzo istotne dla poznania sposobu działania włączników u::Lux jest zrozumienie różnicy między oboma trybami, która dotyczy sposobu prezentowania stanu aktora (sterowanego urządzenia/zmiennej).

### Wartość edytowalna

Poniższy diagram ilustruje aktualizację stanu w trybie pracy aktora jako wartość edytowalna:

## 9. Integracje



W trybie wartości edytowalnej modyfikacji podlega zmienna lokalna w oprogramowaniu włącznika u::Lux. Następnie jej wartość przesyłana jest do modułu Base wraz z numerem aktora. Moduł Base na podstawie utworzonego mapowania, przekazuje otrzymaną wartość jako komendę do przypisanego w konfiguracji identyfikatora i w ten sposób realizowane jest sterowanie. Po wykonaniu się komendy aktualizowany jest stan identyfikatora w Base, a następnie jego nowa wartość jest wysyłana z powrotem do włącznika u::Lux, w celu zapewnienia synchronizacji stanu sterowanego urządzenia/zmiennej. Zapewnia to również prawidłowość wyświetlania aktualnego stanu na ekranie włącznika w przypadku sterowania danym urządzeniem/zmienną przy pomocy innych dostępnych w DOMIQ sposobów, np. przy użyciu aplikacji Remote.

### Wartość rzeczywista

Poniższy diagram ilustruje aktualizację stanu w trybie pracy aktora jako wartość rzeczywista:



Aktor pracujący w trybie wartości rzeczywistej wysyła jedynie informację o naciśnięciu lub zwolnieniu klawisza. Moduł Base przetwarza tę informację wg zaprogramowanych reguł i steruje przypisanym identyfikatorem. Następnie stan sterowanego identyfikatora (urządzenia/zmiennej) jest wysyłany z modułu Base do włącznika/włączników u::Lux w celu aktualizacji stanu na wyświetlaczu.

W module Base dodaliśmy także funkcjonalność umożliwiającą rozróżnianie długości naciśnięcia klawisza. W ten sposób rozróżniane jest krótkie naciśnięcie, długie naciśnięcie (powyżej 1 sekundy) oraz zwolnienie klawisza. Informacja o naciśnięciu klawisza jest odbierana przez moduł Base jako zdarzenie, na które można zareagować dowolną akcją.

Opis zdarzeń generowanych przez interakcję z klawiszami znajduje się w rozdziale 12. Identyfikatory.

W przypadku, gdy mapowanie dotyczy identyfikatora, który posiada implementację funkcji ściemniania, wówczas długie naciśnięcie rozpoczyna płynną zmianę wartości (rozjaśnianie/ściemnianie), a kolejne długie naciśnięcie zmienia kierunek zmian wartości na przeciwny. Puszczenie klawisza zatrzymuje zmianę wartości.

Opcjonalnie istnieje możliwość przypisania komend, które będą wykonywane przy określonym rodzaju interakcji z klawiszem. Wypełnienie poniżej opisanych pól jest opcjonalne (można wypełnić tylko jedno z nich, dwa, wszystkie lub wcale). W przypadku, gdy pola zostaną puste, wówczas naciśnięcie klawisza zostanie obsłużone w sposób domyślny, opisany powyżej.

- **Krótko** – Komenda, która zostanie wykonana przy krótkim (poniżej sekundy) naciśnięciu klawisza. Poniższy przykład jest pasujący do pozostałych rodzajów akcji



- C.LCN.output.0.12.2=100  
Wysterowanie wyjścia nr 2 na 100% w module LCN o adresie 12.

- **Długo** – Komenda, która zostanie wykonana przy długim (powyżej sekundy) naciśnięciu klawisza.
- **Puść** – Komenda, która zostanie wykonana, gdy klawisz zostanie puszczone.

W górnej części interfejsu znajdują się trzy przyciski:

- **Zapisz** – Zapisuje konfigurację zakładki **ULUX**.  
**WAŻNE: utworzone powiązania nie będą aktywne dopóki nie zostanie zapisana konfiguracja.** Wszystkie zmiany wprowadzone w konfiguracji są pamiętane lokalnie w przeglądarce do czasu zapisania konfiguracji.
- **Odtwórz** – Odtwarza ostatnią zapisaną konfigurację zakładki **ULUX**.



- **Skanuj** – rozpoczyna skanowanie w poszukiwaniu włączników u::Lux obecnych w sieci LAN. Skanowanie należy wykonać po ręcznej zmianie adresów IP podłączonych włączników lub po ręcznej zmianie ich identyfikatorów w oprogramowaniu u::Lux Config. Jeżeli nie były wprowadzone żadne zmiany w tym zakresie, wówczas skanowanie nie jest potrzebne. Włączniki nowo podłączone do instalacji zostaną wykryte automatycznie.



### Przykład 1:

W tym przykładzie zaprezentujemy tworzenie mapowania, które umożliwi sterowanie wyjściem ściemnianym LCN w sposób skokowy.

1. Kliknij przycisk **Dodaj**, aby dodać aktora.
2. **ID aktora** - wpisz numer aktora zgodnie z konfiguracją z oprogramowania u::Lux Config.
3. **Kanał** - wpisz identyfikator wyjścia ściemnianego w module LCN, np. LCN.output.0.10.1.
4. Wybierz pole **Wartość edytowalna**.
5. Kliknij przycisk **Zapisz**, aby potwierdzić konfigurację.

Mając wybrany tryb aktora jako Wartość edytowalna (taki sam tryb musi być wybrany w oprogramowaniu uLux::Config). Następnie w oprogramowaniu uLux::Config należy zdefiniować parę klawiszy, w której jeden będzie zwiększał wartość, a drugi ją zmniejszał o określony skok.

### Przykład 2:

W tym przykładzie zaprezentujemy natomiast płynne sterowanie wyjściem ściemnianym LCN.

1. Kliknij przycisk **Dodaj**, aby dodać aktora.
2. **ID aktora** - wpisz numer aktora zgodnie z konfiguracją z oprogramowania u::Lux Config.
3. **Kanał** - wpisz identyfikator wyjścia ściemnianego w module LCN, np. LCN.output.0.10.1.
4. Wybierz pole **Wartość rzeczywista**.
5. Pola opcjonalne (**Krótko**, **Długo**, **Puść**) zostaw puste.
6. Kliknij przycisk **Zapisz**, aby potwierdzić konfigurację.

Mając wybrany tryb aktora jako **Wartość rzeczywista** (taki sam tryb musi być wybrany w oprogramowaniu uLux::Config). Krótkie naciśnięcie spowoduje wysterowanie wyjścia na 100% lub 0% (w zależności od obecnego stanu), natomiast przy długim naciśnięciu wartość będzie zmieniała się płynnie. Zwolnienie klawisza zatrzyma zmianę. Ten rodzaj sterowania można zrealizować z wykorzystaniem jednego klawisza.

# 9.8. Shelly

W DOMIQ jesteśmy zwolennikami kablowych instalacji automatyki budynkowej, które cechują się bardzo dużą niezawodnością transmisji danych.

Są jednak sytuacje, gdzie nie ma już możliwości położenia instalacji kablowej. Wychodząc na przeciw tym potrzebom udostępniamy integrację urządzeń Shelly w DOMIQ.

## Komunikacja

Urządzenia Shelly umożliwiają integrację z użyciem API HTTP oraz MQTT. W związku z tym, że API HTTP wymaga stałego odpytywania o bieżący stan urządzeń, wybraliśmy standard MQTT, który zapewnia w pełni dwukierunkową komunikację.

Na potrzeby integracji z Shelly wykonaliśmy własną implementację brokera MQTT. W obecnej formie nasz broker MQTT spełnia część (nie jest jeszcze w pełni kompatybilny) standardu MQTT 3.1.1. Na tym etapie obsługuje komunikację z QoS równym 0 i wymaga ustawienia opcji Clean Session dla każdego połączenia. Więcej informacji o standardzie MQTT można znaleźć w Internecie.

## Konfiguracja

Pierwszym etapem integracji jest prawidłowa konfiguracja modułów Shelly.

### Ustawienia sieci

Wszystkie moduły Shelly fabrycznie startują w trybie Access Point - zatem każdy moduł tworzy swoją lokalną sieć WiFi. Pierwszym krokiem jest zmiana ustawień sieciowych modułu, aby łączył się on z Twoją lokalną siecią WiFi.

1. Podłącz się do sieci utworzonej przez moduł Shelly.
2. Otwórz interfejs konfiguracyjny modułu - wpisz jego domyślny adres IP (patrz instrukcja obsługi Shelly) w polu adresu przeglądarki.
3. W sekcji **Networks** lub **Internet & Security** wprowadź ustawienia umożliwiające podłączenie modułu do twojej lokalnej sieci WiFi. Zalecamy ustawienie statycznego adresu IP lub utworzenie przypisania adresu IP do adresu MAC karty sieciowej (należy to wykonać w ustawieniach routera).
4. Po zapisaniu ustawień otwórz ponownie interfejs konfiguracyjny Shelly, ale z wykorzystaniem już przypisanego adresu IP w Twojej sieci lokalnej.

### Ustawienia MQTT

Następnym etapem jest skonfigurowanie ustawień serwera MQTT, do którego moduł Shelly będzie się podłączał. Opis konfiguracji dotyczy interfejsu Shelly. Konfiguracja nieco różni się dla urządzeń Shelly 1 i 2 generacji, dlatego opiszemy te procesy osobno.

#### Urządzenia generacji 1

1. Zakładka **Internet & Security** -> **Advanced - Developer Settings**
2. Zaznacz opcję **Enable MQTT**.
3. Uzupełnij następujące pola formularza:
  - **Server** - Wpisz adres IP modułu Base oraz domyślny nr portu dla komunikacji MQTT, czyli 1883. Przykład: **192.168.1.100:1883**
  - Zaznacz opcję **Use custom MQTT prefix** i w polu poniżej wpisz nazwę modułu Shelly. **Nazwa musi być unikalna dla całej instalacji!**
  - Zaznacz opcję **Clean Session**.
  - Upewnij się, że opcja **Max QoS** ma wartość 0.
4. Następnie kliknij przycisk **Save**, aby zapisać konfigurację. Do zatwierdzenia ustawień wymagany jest restart urządzenia, o czym poinformuje Cię interfejs Shelly.
5. Konfiguracja modułu zakończona.

## Urządzenia generacji 2

1. Zakładka **Networks** -> **Mqtt**
2. Włącz opcję **Enable**.
3. Uzupełnij następnie pola formularza:
  - **Connection type:** *MQTT*
  - W następnym polu wpisz nazwę modułu Shelly. **Nazwa musi być unikalna dla całej instalacji!**
  - Włącz opcję **RPC status notifications over MQTT**.
  - W polu tekstowym poniżej wpisz adres IP modułu Base oraz domyślny nr portu dla komunikacji MQTT, czyli 1883. Przykład: *192.168.1.100:1883*
4. Kliknij **Apply** aby zatwierdzić ustawienia.
5. Konfiguracja modułu zakończona

## Pozostałe ustawienia

Na tym etapie moduł Shelly jest gotowy do integracji z DOMIQ/Base. Pozostałe ustawienia modułu np. rodzaj używanych wejść, harmonogramy i inne skonfiguruj wg potrzeb.

## Konfiguracja modułu DOMIQ/Base

W konfiguratorze modułu Base, w zakładce **Ustawienia** włącz opcje: **MQTT** oraz **Shelly**. Następnie zapisz zakładkę **Ustawienia** i zrestartuj moduł.

Po restarcie Base jest gotowy do obsługi urządzeń Shelly. Moduły Shelly automatycznie podłączą się do Base - żadne dodatkowe działania nie są wymagane. Parametry odczytane z modułów Shelly automatycznie pojawią się w zakładce **Stan**.

## Sterowanie i odczyt stanu

Jak już wspomnieliśmy wcześniej, moduły Shelly dzielą się na moduły pierwszej i drugiej generacji. Oprócz różnic sprzętowych, różnią się także protokołem integracyjnym. Protokół modułów generacji 2 został ujednoczony dla wszystkich urządzeń, podczas gdy urządzenia pierwszej generacji dość mocno różnią się między sobą pod względem parametrów, które można odczytać, a także komend, które można wysyłać do modułów. Opis obsługi modułów obu generacji podzieliśmy na opis poszczególnych urządzeń i przedstawiliśmy w rozdziale 12.8 SHELLY.

# 9.9. IFTTT

Serwis IFTTT (ang. **If This Then That**) oferuje możliwość tworzenia zależności logicznych z wykorzystaniem dostępnych w serwisie apletów. Jednym z nich jest aplet **webhooks**. Dzięki niemu DOMIQ został otwarty na całkowicie nowy wymiar integracji. Aplet webhooks pozwala na wysyłanie do serwera IFTTT żądań HTTP, które uruchomią utworzone przez Ciebie aplety. Jest także funkcjonalność odwrotna – aplet może wysłać żądanie HTTP do modułu Base, co pozwala na wywołanie dowolnego zdarzenia w systemie. Jak to przekłada się na funkcjonalność Base? Wymieńmy tylko kilka najciekawszych możliwości, jakie daje integracja z IFTTT, a jest ich naprawdę sporo i wciąż przybywają nowe aplety:

- wywoływanie zdarzeń w Base na podstawie lokalizacji Twojego urządzenia mobilnego.
- integracja z nowymi urządzeniami (np. Philips Hue, Amazon Alexa, termostat Nest i wiele innych).
- Integracja z kontem Google, a dzięki temu np. możliwość eksportowania dowolnych danych dostępnych w module Base do arkusza kalkulacyjnego, wysyłanie maili na podstawie zdarzeń w systemie inteligentnego domu, itd.
- Powiązanie systemu DOMIQ z informacjami pogodowymi.
- Integracja z natywnymi funkcjami systemu iOS (kalendarz, kontakty).
- Integracja z aplikacją Skype – możliwość wysłania wiadomości w reakcji na zdarzenie w systemie DOMIQ.

Opis komend wywołujących aplety IFTTT i zdarzeń na odbieranie żądań przychodzących z serwera IFTTT zostało opisane w podrozdziale 12.10 IFTTT

Szczegółowy opis integracji znajdziesz w samouczku **Integracja z IFTTT**: <https://www.domiq.pl/wiki/integracja-z-ifttt/>.

# 9.10. Grafana i Prometheus.io

Integracja z oprogramowania Prometheus.io i Grafana dają ogromne możliwości w zakresie akwizycji i prezentacji danych, m.in na wykresach, grafach itd. Pełny opis integracji został zaprezentowany w samouczku **Wykresy w Remote** dostępnym pod tym linkiem: <https://www.domiq.pl/wiki/wykresy-w-domiq-integracja-z-grafana/>

Oprogramowanie Grafana umożliwia wywoływanie tzw. alarmów, kiedy obserwowane dane spełniają określone parametry, np. kiedy obserwowana wartość przekroczy próg krytyczny. Integracja z oprogramowaniem Grafana umożliwia, aby moduł Base odbierał notyfikacje o wystąpieniu alarmów, co w systemie DOMIQ jest rozgłaszane w postaci zdarzeń.

Opis tych zdarzeń został zaprezentowany w podrozdziale 12.11 Grafana.

## 9.11. Satel Integra

Integracja z centralami alarmowymi Satel Integra jest jedną z najczęściej wybieranych z grona opcji integracyjnych oferowanych przez system DOMIQ. Integracja z centralami z serii Integra może odbywać się na dwa sposoby:

1. Z użyciem modułu Satel INT-RS Plus/INT-RS za pośrednictwem standardu RS-232.
2. Z użyciem modułu Satel ETHM-1 Plus (firmware 2.05 lub nowszy) lub ETHM-1 (firmware 1.07 lub nowszy) za pośrednictwem protokołu TCP i sieci LAN.

Poniżej omówione zostały obie opcje integracji.

### INT-RS Plus / INT-RS

W tej metodzie integracji system DOMIQ należy wyposażyć w moduł Serial-2SI, który stanowi bramkę sprzęgającą z modułem INT-RS po stronie centrali alarmowej. Komunikacja odbywa się z użyciem standardu RS-232.

Dostępna funkcjonalność:

- **Stan wejść/wyjść i stref alarmowych.**
- **Sterowanie wyjściami centrali alarmowej.**
- **Uzbrajanie/rozbrajanie stref alarmowych.**

Wadą tej metody integracji są opóźnienia w przesyłaniu danych wynikające ze sposobu komunikacji. Opóźnienia mogą sięgać nawet do 5s!

### ETHM-1 Plus/ETHM-1

W tej metodzie integracji w systemie DOMIQ nie jest potrzebne żadne dodatkowe urządzenie. Całą obsługę komunikacji z centralą wykonuje moduł Base, do którego należy dokupić licencję rozszerzającą go o możliwości integracyjne z Satel Integra.

Komunikacja odbywa się z użyciem protokołu TCP i sieci LAN, co eliminuje opóźnienia w przesyłaniu danych, które są charakterystyczne dla integracji z użyciem modułów INT-RS i standardu RS-232. Dzięki temu komendy sterujące oraz statusy zwrotne są dostarczane błyskawicznie (jest to realnie obserwowalne).

Możliwości integracyjne zostały ponadto poszerzone o:

- **Odczyt logów zdarzeń z centrali alarmowej.**
  - Logi są pobierane automatycznie co minutę.
  - Możliwość wyboru zakresu czasowego - od 1 do 7 dni. Domyślnie moduł Base odczytuje zdarzenia dla ostatnich dwóch dni.
  - Automatycznie generowany jest log zbiorczy dla zdarzeń z bieżącego dnia (ostatnich 20 wpisów). Dzięki temu z łatwością wyświetlisz najświeższe logi o zdarzeniach w aplikacji Remote.
- **Odczyt informacji o awariach (grupa nr 4 notyfikacji o problemach)**
  - brak komunikacji z ekspanderem
  - podmiana ekspandera
  - brak komunikacji z manipulatorem
  - podmiana manipulatora
  - brak kabla w module ETHM-1
  - sabotaż manipulatorów
  - sabotaż ekspanderów
  - błąd inicjalizacji manipulatora

Komendy, zdarzenia i identyfikatory stanu centrali zostały opisane w rozdziale 12.3 IDS



# Rozdział 10

## Łącza

Zakładka **Łącza** służy do definiowania połączeń pomiędzy modułami Base. Połączenia tego typu zostały zaprojektowane z myślą o rozbudowanych instalacjach automatyki budynkowej, gdzie potrzebne jest zastosowanie więcej niż jednego modułu Base.

Interfejs podzielony jest na trzy elementy:

1. **Struktura** - W niej wyświetlane są wszystkie zdefiniowane połączenia. Nad tabelą znajdują się przyciski służące do dodawania/usuwania połączeń.
2. **Szczegóły** - Ten panel służy do konfiguracji parametrów danego połączenia. Jego opcje zostały opisane w dalszej części tego rozdziału.
3. **Dostępne moduły** - W tym oknie są wyświetlane moduły Base podłączone do tej samej sieci LAN.
  - **Nazwa** - Nazwa nadana modułowi Base w zakładce **Ustawienia**.
  - **Adres** - Adres IP modułu.
  - **Typ** - Typ modułu.
  - **Wersja** - Aktualna wersja oprogramowania modułu.
  - **Identyfikator** - Unikalny identyfikator modułu Base.

Przycisk **Zapisz** zapisuje bieżącą konfigurację w pamięci modułu Base, natomiast przycisk **Przywróć** otwiera ostatnio zapisaną konfigurację zakładki.

### Konfiguracja połączeń

#### 1. Połączenie

Jest to rodzaj połączeń typu punkt-punkt, który łączy ze sobą dwa moduły Base i umożliwia im wzajemną interakcję poprzez komendy i zdarzenia sieciowe. Zarezerwowane zostały do tego specjalne sieciowe identyfikatory. Opis składni identyfikatorów z przykładami znajdziesz w podrozdziale 12.16 „Łącza”.

Moduły Base mogą łączyć się zarówno przy użyciu sieci lokalnej jak i internetu. **W przypadku połączeń internetowych konieczne jest otwarcie portu UDP 44544.**

Doskonałym obszarem zastosowań tego typu połączeń są np. budynki wielomieszkaniowe lub hotele. W takich obiektach często tworzona jest sieć hierarchiczna, gdzie jeden z modułów Base pełni rolę modułu nadrzędnego – przetwarza informacje otrzymane z modułów podrzędnych (np. zamontowanych w mieszkaniach).

Połączenie tego typu wymaga konfiguracji krzyżowej, czyli: w konfiguracji modułu A wpisujesz dane modułu B, a w konfiguracji modułu B dane modułu A.

W panelu **Szczegóły** dostępne są następujące parametry:

- **Nazwa** (pole wymagane) - Nazwa modułu Base, z którym chcesz nawiązać połączenie. Nazwa może być dowolna bez polskich znaków i spacji. Wpisana tutaj nazwa nie ma związku z nazwą modułu nadaną w zakładce **Ustawienia**.
- **Opis** - Krótki opis urządzenia.
- **Adres** (pole wymagane) - Adres IP modułu, z którym chcesz nawiązać połączenie.
- **Port** (opcjonalne) - Numer portu, na którym ma być nawiązane połączenie. Uzupełnij to pole tylko wtedy, gdy chcesz użyć niestandardowego portu. Domyślnie wewnątrz używany jest port UDP o numerze 44544.

## 10. Łącza

- **Hasło:** Jeżeli to pole zostanie uzupełnione, wówczas dane przesyłane pomiędzy modułami będą szyfrowane.
- **Identyfikator:** Każdy moduł Base ma swój unikalny identyfikator. Identyfikator pełni funkcję dodatkowego zabezpieczenia transmisji danych pomiędzy modułami Base. Uniemożliwia on wykorzystanie adresu IP przypisanego do modułu Base posługującego się danym identyfikatorem przez inne urządzenie sieciowe. Identyfikatory widoczne są w tabeli **Dostępne moduły**.



Konfiguracja pojedynczego połączenia wykonywana jest „na krzyż”. W konfiguracji modułu A, w polach **Nazwa** oraz **Adres** podajemy dane modułu, z którym chcesz nawiązać połączenie i odwrotnie. Załóżmy, że moduły wykorzystują kolejno adresy 192.168.1.100 oraz 192.168.1.101 oraz mają nadane nazwy `domiq1` oraz `domiq2`. Konfiguracja przebiega wg poniższej procedury:

### Moduł A (`domiq1`)

- **Nazwa:** `domiq2`
- **Adres:** `192.168.1.101`.
- Jeżeli połączenie ma być szyfrowane uzupełnij pole **Hasło**
- Opcjonalnie uzupełnij komórki **Identyfikator**, **Opis** oraz **Port**.

### Moduł B (`domiq2`)

Powtórz kroki 1 i 2, pamiętając o zmianie adresu IP i nazwy na dane modułu pierwszego, zatem: 192.168.1.100 oraz `domiq1`.

Więcej o połączeniach punkt-punkt przeczytasz w samouczku: <https://www.domiq.pl/wiki/laczenie-wielu-modulow-base/>

## 2. Grupa

Grupa służy do łączenia dwóch lub więcej modułów Base we współpracującą ze sobą grupę. Wewnątrz grupy moduły mogą wymieniać ze sobą dane przez współdzielone zmienne sieciowe. W takiej instalacji jeden z modułów Base może (nie jest to wymagane) pełnić rolę nadrzedną i dostarczać informacji innym modułom (za pomocą zmiennych sieciowych) np. o sytuacjach alarmowych (pożar, zalanie, itp.). Wymiana zmiennych sieciowych jest dwukierunkowa pomiędzy wszystkimi modułami przynależnymi do danej grupy: wszystkie moduły w obrębie danej grupy mają dostęp do stanu danej zmiennej sieciowej, mogą zmieniać ten stan przy użyciu komend oraz reagować na zmiany stanu przy użyciu zdarzeń.

Wartości zmiennych sieciowych są przechowywane równolegle we wszystkich modułach Base należących do danej grupy. Restart modułu Base powoduje skasowanie zmiennych z pamięci. Jednakże po ponownym uruchomieniu, moduł Base automatycznie synchronizuje zmienne poprzez pobranie ich wartości z innego modułu przynależnego do grupy. Wartości zmiennych sieciowych zostaną całkowicie skasowane jedynie w przypadku jednoczesnego zrestartowania wszystkich modułów Base przynależnych do grupy.

Po dodaniu nowej grupy w zakładce **Łącza** należy skonfigurować jej parametry:

- W polu **Nazwa** należy wpisać nazwę grupy. Nazwę tą należy wprowadzić we wszystkich modułach Base, które mają przynależeć do danej grupy.
- Pozostałe pola są opcjonalne.

Komunikacja pomiędzy modułami w grupie odbywa się z użyciem mechanizmu multicast. W multicast zgodnie z normą RFC 3171 adresy IP nadawane są w zakresie od 224.0.0.0 do 239.255.255.255. Moduły Base domyślnie mają ustawiony adres IP multicast `239.255.255.44` oraz port: `44544`. Jeżeli chcesz, aby grupa komunikowała się z użyciem domyślnego adresu IP multicast, wówczas pole **Adres** może pozostać puste. W przeciwnym przypadku należy je uzupełnić innym adresem IP z zakresu podanego powyżej. Czynność należy wykonać dla wszystkich modułów Base należących do danej grupy.

Do obsługi zmiennych sieciowych używane są dedykowane identyfikatory typu NET. Dokładny ich opis znajdziesz w podrozdziale: 12.2 NET.



## Base jako sprzęg segmentowy LCN

Jest to szczególny przykład użycia grupy, który umożliwia wykorzystanie modułów Base, jako sprzęgów segmentowych w instalacji LCN.

**Nazwa:** LCN

**Każda inna nazwa uniemożliwi komunikację pomiędzy segmentami LCN.**

**Adres:** Mają tu zastosowania te same zasady, jak opisane powyżej.

Ostatnim krokiem jest przypisanie modułu Base do określonego segmentu LCN.

1. Przejdź do zakładki **Ustawienia** -> **Integracje**, w polu **Segment** wpisz numer segmentu LCN.
2. Włącz opcję **Zdarzenia z innych segmentów**, aby moduł Base mógł odbierać i reagować na zdarzenia pochodzące z innych segmentów LCN. Zaznaczenie tego pola oznacza jednocześnie, że moduł Base będzie przechowywał w swoim stanie stan identyfikatorów z innych modułów Base, pełniących rolę sprzęgu segmentowego.

Komendy i zdarzenia pochodzące z innych segmentów mają standardową składnię. Różnią się jedynie numerem segmentu w treści identyfikatora, zgodnie z przyjętymi ustawieniami.



- `E.LCN.output.10.11.1=100`  
Wyjście ściemniacze nr 1 w module o adresie 11, w segmencie nr 10 zostało włączone na 100%.
- `E.LCN.relay.10.11.5=0`  
Przełącznik nr 5 w module nr 11, w segmencie 10 został wyłączony.
- `C.LCN.output.10.11.1=100`  
Wysteruj na 100% wyjście ściemniacze nr 1 w module o adresie 11, w segmencie 10.

Więcej szczegółowych informacji nt. użycia modułu Base jako sprzęgu segmentowego znajdziesz w samouczku pt. „**Base jako sprzęg segmentowy**”: <https://www.domiq.pl/wiki/base-jako-sprzeg-segmentowy-lcn/>

## 3. Master i Slave

Te dwa rodzaje połączeń umożliwiają tworzenie hierarchii modułów Base, gdzie jeden z nich pełni rolę nadrzędną (master), a drugi podrzędną (slave). Miejscem zastosowania tego typu połączeń są instalacje, gdzie pracują co najmniej dwa moduły DOMIQ/Base i niezbędne jest sterowanie nimi z użyciem wspólnego interfejsu użytkownika.

### Właściwości połączeń master-slave:

- Moduł nadrzędny ma dostęp do stanu wszystkich urządzeń i zmiennych dostępnych w module podrzędnym.
- Moduł nadrzędny otrzymuje informacje o zmianach stanu w module podrzędnym.
- Moduł nadrzędny ma dostęp do funkcji w zakładce **Logika** modułu podrzędnego.
- Moduł nadrzędny może wysyłać dowolne komendy do modułu podrzędnego.
- Jeden moduł Base może pełnić rolę nadrzędną dla kilku modułów Base.
- Moduł Base pracujący jako moduł nadrzędny może być jednocześnie modulem podrzędnym względem innego modułu Base. Jednakże zmiany stanu są przekazywane wyłącznie o jeden poziom w górę hierarchii.

**Przykład:** Przyjmijmy, że w instalacji są trzy moduły Base o nazwach kolejno: Base1, Base2 oraz Base3. Moduł Base2 jest nadrzędny względem modułu Base1 i jednocześnie podrzędny względem modułu Base3. W takiej konfiguracji moduł Base3 otrzyma wyłącznie informacje o zmianach stanu identyfikatorów z modułu Base2. Zmiany stanu pochodzące z modułu Base1, wynikające z istnienia połączenia master-slave pomiędzy modułami Base1 i Base2 zostaną w takim przypadku odfiltrowane i nie trafią do modułu Base3.

- Moduł nadrzędny wykona automatyczną synchronizację stanu przypisanego modułu podrzędnego w przypadku, gdy moduł nadrzędny zostanie zrestartowany lub podłączony ponownie do sieci LAN po przerwie dłuższej niż 30s.

## 10. Łącza

- Analogicznie, moduł nadrzędny wykona automatyczną synchronizację stanu przypisanego modułu podrzędnego w przypadku, gdy moduł podrzędny zostanie zrestartowany lub podłączony ponownie do sieci LAN po przerwie dłuższej niż 30s.
- W przypadku odłączenia od sieci LAN na okresy krótsze niż 30s, synchronizacja stanu nastąpi po pierwszej zmianie stanu w module podrzędnym.
- Nie należy tworzyć w module nadrzędnym zdarzeń, które wyzwalane są przez moduł podrzędny zmianami stanu na wartość 0. W przypadku takich zdarzeń będzie dochodzić do ich fałszywego wyzwalania w czasie restartu modułu podrzędnego, gdyż wówczas następuje tymczasowe wyzerowanie stanu.

Dzięki powyższym właściwościom możliwe staje się utworzenie jednego wspólnego interfejsu użytkownika (wizualizacji i/lub menu Remote) dla wszystkich modułów Base pracujących w danej instalacji. Ponadto moduł nadrzędny może wysyłać komendy do modułów podrzędnych w reakcji na zdarzenia, jako efekt działania timera, funkcji w **Logice** itd. Uogólniając, połączenia master-slave umożliwiają sterowanie modułem podrzędnym w sposób taki, jak gdyby wszystko odbywało się w obrębie jednego modułu Base.

### Przykład konfiguracji



#### Moduł nadrzędny

1. Dodaj połączenie z modułem podrzędnym poprzez kliknięcie przycisku **+Slave**.
2. W polu **Nazwa** wpisz nazwę połączenia. Nazwa może składać się wyłącznie z liter od a do z (również wielkich), cyfr oraz znaku podkreślenia. W nazwie nie wolno używać polskich znaków. **Nazwa połączenia musi być koniecznie identyczna po obu jego stronach.**
3. W polu Adres należy wpisać adres IP podrzędnego modułu Base.
4. Pozostałe pola są opcjonalne:
  - W polu **Opis** możesz wpisać pomocniczy opis połączenia.
  - W polu **Port** wpisz numer portu, który będzie używany do nawiązania połączenia. To pole należy modyfikować tylko, gdy jest to uzasadnione np. wynika ze specyficznych ustawień sieci lokalnej.
  - W polu **Hasło** możesz wpisać hasło, jeżeli chcesz, aby komunikacja była szyfrowana. Z uwagi na fakt, że moduły biorące udział w połączeniu pracują w tej samej sieci LAN szyfrowanie można pominąć. Szyfrowanie nieco spowalnia komunikację między modułami Base.
5. Zapisz zakładkę **Łącza**

#### Moduł podrzędny

Konfiguracja w przypadku modułu podrzędnego jest analogiczna.

1. Dodaj połączenie z modułem nadrzędnym klikając przycisk **+Master**.
2. W polu **Adres** wpisz adres IP modułu nadrzędnego.
3. Pozostałe czynności należy wykonać identycznie, jak dla modułu nadrzędnego.

Do obsługi połączeń master-slave używane są identyfikatory MNT. Dokładny ich opis znajdziesz w podrozdziale: 12.16 Łącza.

Ponadto zachęcamy do lektury samouczka o połączeniach master-slave: <https://www.domiq.pl/wiki/polaczenia-master-slave-miedzy-modulami-domiq-base/>

# Rozdział 11

## Status

W zakładce **Status** wyświetlane są informacje diagnostyczne o stanie urządzenia oraz poszczególnych komponentach systemu operacyjnego modułu DOMIQ/Base. Informacje zawarte w zakładce **Status** mogą ułatwić w znacznym stopniu diagnozę nieprawidłowości w działaniu modułu DOMIQ/Base, mogących być skutkiem np. błędów w obsłudze zdarzeń, programie w **Logice** etc

Zakładka zawiera takie informacje jak:

- Całościowe oraz cząstkowe wykorzystanie procesora. Jeżeli któryś proces zużywa dużą część zasobów procesora podczas normalnej pracy, może to oznaczać jego niewłaściwe działanie.
- Informacje o zużyciu i dostępnej pamięci dla każdego procesu.



# Rozdział 12

# Identyfikatory

W tym rozdziale opisane zostały wszystkie identyfikatory dostępne w systemie DOMIQ wraz z przykładami ich zastosowania. Dla przypomnienia - identyfikator to ciąg znaków, jednoznacznie identyfikujący urządzenie lub zmienną. Moduł Base posiada całą gamę predefiniowanych identyfikatorów oraz umożliwia użytkownikom definiowanie własnych identyfikatorów. Przy użyciu identyfikatora możesz wysłać komendę, zareagować na zdarzenia dotyczące danego identyfikatora oraz wyświetlić odczytać jego aktualny stan.

# 12.1. Zmienne

Identyfikatory z tej grupy nie odnoszą się do konkretnych urządzeń, wskazują zaś na zmienne przechowywane w pamięci modułu Base. Część zmiennych umożliwia zapis i odczyt, a część z nich jest tylko do odczytu. Identyfikatory z tej grupy znajdują zastosowanie zwłaszcza przy definiowaniu reguł czasowych, zdarzeń warunkowych, a także tworzeniu skryptów w zakładce **Logika**.

## MEM

Identyfikatory tego typu wskazują na zmienne, które zapisywane są w pamięci nieulotnej modułu Base. Zmienne typu MEM zachowują swoją wartość nawet po zaniku zasilania.

Zmienne MEM znajdują szerokie zastosowanie przy definiowaniu timerów, zdarzeń warunkowych oraz tych elementów interfejsów użytkownika, których nastawy powinny być trwale zapamiętywane.

	Identyfikator	Wartość	Opis
<b>Stan</b>	MEM.nazwa	dowolna	Aktualny stan zmiennej MEM.
<b>Zdarzenie</b>	E.MEM.nazwa	dowolna	Informacja o zmianie stanu zmiennej.
<b>Komenda</b>	C.MEM.nazwa	dowolna	Zapisanie dowolnej wartości w zmiennej.



### 1. Użycie stanu zmiennej MEM jako warunek wykonania akcji timera.

Załóżmy, że budynek wyposażony jest system podlewania ogrodu, który uruchamiany jest o określonych godzinach w ciągu dnia. Jednakże dodatkowo użytkownik chce mieć możliwość ręcznej deaktywacji automatycznego uruchomienia tego procesu. Najłatwiej jest to osiągnąć dodając element typu **Przełącznik** w zakładce **Remote** i przypisując do niego zmienną typu MEM. Wówczas stan przełącznika (przyjmuje wartości `on/off`) będzie zapisany w zmiennej nieulotnej, a jego wartość można wykorzystać w warunku wykonania akcji przypisanych do timera.

Załóżmy, że w polu **Kanał** przełącznika wpisano: `MEM.autopodlewanie`, wówczas w polu **Warunek** w definicji timera należy wpisać: `MEM.autopodlewanie == "on"`. W ten sposób stan zmiennej MEM determinuje, czy timer zostanie wykonany.

### 2. Zdarzenie dla zmiany stanu zmiennej MEM

Każda zmiana stanu identyfikatora w systemie DOMIQ generuje zdarzenie o tym informujące. Dla każdej zmiany stanu zmiennej MEM opisanej w poprzednim przykładzie zostanie rozgłoszone zdarzenie, które przekazywać będzie nową wartość zmiennej.

Wówczas bardzo łatwo jest utworzyć regułę zdarzenia, które będzie aktywowane wtedy, gdy zmienna osiągnie interesując nas wartość.

Jeżeli zmienna `MEM.autopodlewanie` osiągnie wartość `on`, wówczas rozgłoszone zostanie zdarzenie `E.MEM.autopodlewanie` z wartością `on`.

### 3. Komenda zmieniająca stan zmiennej MEM.

Rozbudowując nasz przykład automatycznego podlewania, załóżmy że system ponadto wyposażony jest w czujnik deszczu i towarzyszącą temu regułę, aby automatyczne podlewanie było blokowane, jeżeli opady deszczu były np. na określonym poziomie lub np. trwały przez określony czas. Wówczas efektem działania takiej reguły może być wysłanie komendy zmieniającej stan zmiennej `MEM.autopodlewanie`, aby zablokować wykonanie timera. Każda komenda powinna być poprzedzona prefiksem `C.`

Jeżeli komenda znajduje się w sekcji **Akcje** w zdarzeniach, wówczas w polu **Nazwa** należy wpisać `C.MEM.autopodlewanie`, a w polu **Wartość**: `off`.

## VAR

Zmienne VAR mają identyczne działanie jak zmienne MEM, jednak ich wartości znikają po restarcie modułu.

## Zmienne systemowe

Jest to zbiór identyfikatorów, które reprezentują zmienne udostępniane przez system operacyjny modułu Base. Zmienne tego typu są tylko do odczytu.

Nazwa identyfikatora	Opis identyfikatora	Stan	Zdarzenia	Komenda
CLOCK.date	Aktualna data w formacie DD.MM.RRRR	+	+	-
CLOCK.datetime	Aktualna data i czas w formacie DD.MM.RRRR HH:MM.	+	+	-
CLOCK.day	Numer dnia w miesiącu.	+	+	-
CLOCK.dayname	Skrócona nazwa dnia tygodnia.	+	+	-
CLOCK.days	Liczba dni w danym miesiącu. Szczególnie przydatne, gdy potrzebne jest utworzenie timera, który ma wykonać się w ostatni dzień miesiąca.	+	+	-
CLOCK.fulldate	Aktualna data i godzina wraz z nazwą dnia w formacie DD.MM.RRRR HH:MM.	+	+	-
CLOCK.hour	Aktualna godzina.	+	+	-
CLOCK.minute	Aktualna minuta.	+	+	-
CLOCK.month	Numer miesiąca w roku.	+	+	-
CLOCK.night	Zmienna przyjmuje wartość 1, gdy jest po zachodzie słońca. Wartość jest zerowana o wschodzie słońca.	+	+	-
CLOCK.sunrise	Godzina wschodu słońca w formacie HH:MM	+	+	-
CLOCK.sunset	Godzina zachodu słońca w formacie HH:MM	+	+	-
CLOCK.weekday	Numer dnia w tygodniu, 1 - niedziela, 7 - sobota	+	+	-
CLOCK.time	Aktualny czas w formacie HH:MM	+	+	-
DIQ.controloverinternet	Zmienna informująca o aktywowaniu opcji sterowania przez internet w zakładce <b>Ustawienia</b> .	+	+	-
SUN.azimuth	Azymut słońca na horyzoncie. Może znaleźć zastosowanie do sterowania roletami/żaluzjami w powiązaniu z pozycją słońca na niebie.	+	+	-
SUN.elevation	Kąt słońca nad horyzontem. Zmienna osiąga wartości ujemne po zachodzie słońca.	+	+	-

## 12.2. LCN

Identyfikatory z grupy **LCN** są dedykowane do zastosowania z urządzeniami systemu LCN. Każdy identyfikator realizuje ściśle określoną funkcjonalność powiązaną z systemem LCN.

Poniższa tabela zawiera przegląd wszystkich identyfikatorów **LCN** i ich główne cechy. Szczegółowy opis poszczególnych identyfikatorów wraz z przykładami zastosowania znajduje się w dalszej części podrozdziału.

### Komendy indywidualne i grupowe

Wszystkie opisane poniżej identyfikatory można stosować zarówno do pojedynczych modułów LCN, jak i do grup modułów. Zachodzi tu jednak pewna różnica w składni.

Dla przejrzystości opisu w dalszej części rozdziału przedstawiane będą tylko przykłady komend indywidualnych.

Zalecane jest używanie komend grupowych tak często, jak to możliwe, gdyż komendy grupowe znacznie redukują ruch na magistrali LCN. W przypadku, gdy potrzebujesz wysłać jednocześnie komendę do wielu modułów LCN, rozważ w pierwszej kolejności połączenie ich w grupę.



- **Komenda indywidualna**

`C.LCN.output.0.10.1=on` - przykład komendy załączającej wyjście ściemniane nr 1 w module o adresie 10.

- **Komenda do grupy**

`C.LCN.output.0g10.1=on` - przykład komendy załączającej wyjście ściemniane nr 1 w modułach należących do grupy nr 10.

- **Komenda do wszystkich modułów**

W systemie LCN grupa nr 3 oznacza wszystkie moduły w instalacji. Dzięki temu bardzo łatwo jest wysłać określony typ komendy do wszystkich modułów w instalacji jednocześnie.

`C.LCN.output.0g3.1=on` - komenda załączająca wyjście ściemniane nr 1 we wszystkich modułach LCN.

Nazwa identyfikatora	Skrócony opis	Stan	Zdarzenia	Komenda
<code>LCN.output</code>	Sterowanie i stan pojedynczego wyjścia ściemnianego.	+	+	+
<code>LCN.outputs</code>	Jednoczesne sterowanie wielu wyjść ściemnianych.	-	-	+
<code>LCN.relay</code>	Sterowanie i stan pojedynczego przekaźnika	+	+	+
<code>LCN.relays</code>	Sterowanie kilkoma przekaźnikami.	-	-	+
<code>LCN.sensor</code>	Stan czujnika LCN.	+	+	-
<code>LCN.motor</code>	Sterowanie i stan pojedynczej rolety.	+	+	+
<code>LCN.motors</code>	Jednoczesne sterowanie kilku rolet.	-	+	+
<code>LCN.regulator</code>	Sterowanie i informacje o zmianach stanu regulatora LCN.	+	+	+
<code>LCN.value</code>	Stan i informacje o zmianach wartości temperatury mierzonej przez czujnik LCN.	+	+	-
<code>LCN.variable</code>	Stan i zdarzenia informujące o zmianie wartości jednej z dwunastu zmiennych LCN.	+	+	-
<code>LCN.threshold</code>	Sterowanie wartościami progowymi	+	+	+



LCN.key	Zdarzenie informujące o naciśnięciu przycisku LCN.	-	+	-
LCN.sendkey	Komenda wysyłająca klawisze do systemu LCN.	-	-	+
LCN.scene	Zdarzenie informujące o wywołaniu sceny w LCN.	-	+	+
LCN.scenes	Zapis/odczyt scen świetlnych	-	-	+
LCN.transponder	Zdarzenie informujące o naciśnięciu przycisku na transponderze LCN.	-	+	-
LCN.dali	Komendy sterujące oświetleniem DALI podłączonym do modułu LCN.	-	-	+
LCN.locks	Komenda blokująca klawisze w wybranej tablicy	-	-	+
LCN.text	Komenda umożliwiająca wyświetlanie dowolnego tekstu na wyświetlaczach przycisków z serii GT-D.	-	-	+
LCN.groups	Komendy umożliwiające zarządzanie grupami LCN	-	-	+

## 12. Identyfikatory

### LCN.output

Identyfikator `LCN.output` służy do zarządzania pojedynczym wyjściem ściemnianym modułu LCN. Umożliwia sterowanie, odczyt bieżącej wartości oraz odbieranie zdarzeń o zmianie stanu.

	Identyfikator	Wartość	Opis
<b>Stan</b>	<code>LCN.output.S.M.O</code> s - Segment m - Moduł o - Wyjście (od 1 do 4)	0-100	Procentowa wartość wyjścia ściemnianego
<b>Zdarzenie</b>	<code>E.LCN.output.S.M.O</code>	0-100	Informacja o zmianie stanu wyjścia.
<b>Komenda</b>	<code>C.LCN.output.S.M.O</code>	0-100	Ustaw wskazaną wartość.
		on	Włącz
		off	Wyłącz
		<code>&lt;komenda&gt;;ramp:&lt;rampa&gt;</code>	Wszystkie powyższe komendy z opcją użycia rampy. Rampa określa czas przejścia ze stanu obecnego do zadane-go.
		<code>change:xx</code>	Procentowa zmiana jasności względem wartości aktualnej. Podanie wartości ujemnej oznacza ściemnianie.
		toggle	Przełącz stan na przeciwny.
	stop	Zatrzymuje zmianę jasności wywołaną przez komendę z rampą.	



## 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach.

- **Włączenie wyjścia ściemnianego na max. wartość:**

**Nazwa:** C.LCN.output.0.10.1

**Wartość:** on

Włącz wyjście ściemniane nr 1 w module LCN o adresie 10.

- **Ustawienie określonej jasności z rampą:**

**Nazwa:** C.LCN.output.0.10.1

**Wartość:** 30 ; ramp : 10

Ustaw wyjście nr 1 w module z adresem 10 na wartość 30%, użyj rampy o wartości 10.

- **Relatywna zmiana jasności:**

**Nazwa:** C.LCN.output.0.10.1

**Wartość:** change : 30

Zwiększ jasność wyjścia nr 1 w module z adresem 10 o 30%, .

- **Przełączenie stanu na przeciwny:**

**Nazwa:** C.LCN.output.0.10.1

**Wartość:** toggle

## 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie reagujące na wysterowanie wyjścia na 100%

**Kanał:** E.LCN.output.0.10.1

**Dane:** 100

- Zdarzenie wyzwalane dowolną zmianą wartości wyjścia:

**Kanał:** E.LCN.output.0.10.1

**Dane:** pozostaw to pole puste

Wartość z jaką zostało wywołane zdarzenie dostępna jest pod parametrem \$D0

**UWAGA:** W przypadku modułów z zaprogramowanym ściemnianiem 200 poziomowym, niezbędne jest zaznaczenie opcji **Tryb 200 poziomów** w zakładce **Ustawienia**. W przeciwnym wypadku nieprawidłowo będą realizowane komendy oraz wyświetlany stan wyjść (objawem jest wysterowanie do wartości 25%). W sytuacji, gdy w instalacji występują moduły starszej generacji (nie posiadające 200 poziomowego ściemniania), wówczas należy przeprogramować całą instalację na ściemnianie 50 poziomowe.

## 12. Identyfikatory

### LCN.outputs

Identyfikator `LCN.outputs` umożliwia jednoczesne sterowanie więcej niż jednym wyjściem ściemnianymi w obrębie pojedynczego modułu LCN.

	Identyfikator	Wartość	Opis
Komenda	<b>C</b> .LCN.outputs. <b>S.M.O</b> <b>s</b> - Segment <b>m</b> - Moduł <b>o</b> - Numery wyjść, które mają zostać wysterylowane podane jako przedział, np. 1-4, 2-3, 3-4, 2-4 itd.	toggle	Przełącz na przeciwny
		on	Włącz
		off	Wyłącz



#### Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach.

- **Nazwa:** `C.LCN.outputs.0.10.1-4`  
**Wartość:** `on;ramp:5`  
Włącz wszystkie wyjścia w module o adresie 10 z rampą równą 5.
- **Nazwa:** `C.LCN.outputs.0.10.2-3`  
**Wartość:** `toggle`  
Przełącz wyjścia 2 i 3 na stan przeciwny.

## LCN.relay

Identyfikator `LCN.relay` służy do zarządzania pojedynczym przekaźnikiem w module LCN. Umożliwia sterowanie, odczyt bieżącej wartości oraz odbieranie zdarzeń o zmianie stanu.

	Identyfikator	Wartość	Opis
<b>Stan</b>	<code>LCN.relay.S.M.P</code> S - Segment	0	Przełącznik wyłączony
	<code>M - Moduł</code> <code>P - Przełącznik</code>	1	Przełącznik włączony
<b>Zdarzenie</b>	<code>E.LCN.relay.S.M.P</code>	0	Przełącznik został wyłączony
		1	Przełącznik został włączony
<b>Komenda</b>	<code>C.LCN.relay.S.M.P</code>	0/off	Wyłącz
		toggle	Przełącz na stan przeciwny
		1/on	Włącz
		1;timer:xx	Włączenie przełącznika na określony czas.



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach.

- Włączenie przełącznika**  
**Nazwa:** `C.LCN.relay.0.10.3`  
**Wartość:** `on`  
 Włącz przełącznik nr 3 w module LCN o adresie 10.
- Przełączenie przełącznika**  
**Nazwa:** `C.LCN.relay.0.10.3`  
**Wartość:** `toggle`  
 Przełącz przełącznik nr 3 w module LCN o adresie 10.

### 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie reagujące na wyłączenie przełącznika  
**Kanał:** `E.LCN.relay.0.10.1`  
**Dane:** `off`
- Zdarzenie wyzwalane dowolną zmianą wartości przełącznika:  
**Kanał:** `E.LCN.relay.0.10.1`  
**Dane:** pozostaw to pole puste  
 Wartość z jaką zostało wywołane zdarzenie dostępna jest pod parametrem `$D0`

## 12. Identyfikatory

### LCN.relays

Identyfikator `LCN.relays` wykorzystywany jest do sterowania jednocześnie maksymalnie ośmioma przełącznikami LCN.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<code>C.LCN.relays.S.M</code> S - Segment M - Moduł	<code>query</code>	Odpytanie o aktualny stan przełączników.
		<code>0/1/T/-</code>	Komenda składająca się z 8 znaków, gdzie poszczególne znaki odpowiadają kolejnym przełącznikom. 0 - wyłącz przełącznik 1 - włącz przełącznik T - zmień stan przełącznika na przeciwny - - nie zmieniaj stanu
		<code>&lt;stan_przełączników&gt;;timer:xx</code>	Komenda ustawiająca stan przełączników na określony interwał czasowy. <code>&lt;stan_przełączników&gt;</code> należy używać, jak w komendzie powyżej. Timer wyrażony w milisekundach.



Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach.

- **Dane:** `C.LCN.relays.0.10`  
**Wartość:** `--11----`  
Włącz przełączniki nr 3 i 4 w module LCN o adresie 10. Pozostałe przełączniki pozostaw bez zmian.
- **Dane:** `C.LCN.relays.0.1`  
**Wartość:** `TT1100--`  
Przełącz na wartość przeciwną dwa pierwsze przełączniki, kolejne dwa włącz, przełączniki nr 5 i 6 wyłącz, ostatnie dwa pozostaw bez zmian, moduł LCN o adresie 1.
- **Dane:** `C.LCN.relays.0.1`  
**Wartość:** `TT1100--;timer:300`  
Modyfikacja drugiego przykładu. Komenda ustawi określony stan przełączników na 300ms: przełącz na wartość przeciwną dwa pierwsze przełączniki, kolejne dwa włącz, przełączniki nr 5 i 6 wyłącz, ostatnie dwa pozostaw bez zmian, moduł LCN o adresie 1.

## LCN.sensor

Identyfikator `LCN.sensor` służy do wyświetlania stanu i informacji o zmianach wejść binarnych LCN.

	Identyfikator	Wartość	Opis
<b>Stan</b>	<code>LCN.sensor.S.M.N</code> <b>S</b> - Segment <b>M</b> - Moduł <b>N</b> - Wejście binarne: 1 do 8	0	Wejście nieaktywne
		1	Wejście aktywne
<b>Zdarzenia</b>	<code>E.LCN.sensor.S.M.N</code>	0	Wejście zmieniło się na nieaktywne
		1	Wejście zmieniło się na aktywne



- **Zdarzenie na zmianę stanu**

**Kanał:** `E.LCN.sensor.0.12.2`

**Dane:** 1

Zdarzenie zostanie wywołane, gdy wejście binarne nr 2 w module LCN o adresie 12 zmieni wartość na 1.

- **Odczyt stanu**

`LCN.sensor.0.10.5`

Zwraca stan wejścia binarnego nr 5 modułu LCN o adresie 10.

## 12. Identyfikatory

### LCN.motor

Identyfikator `LCN.motor` używany jest do zarządzania pojedynczą roletą podłączoną do modułu LCN. Rekomendowane jest zastosowanie tego identyfikatora do rolet z pozycjonowaniem. W przypadku rolet bez pozycjonowania należy używać identyfikatora `LCN.relays`.

	Identyfikator	Wartość	Opis
<b>Stan</b>	<code>LCN.motor.S.M.R</code> S - Segment M - Moduł R - Roleta	0-200	0 - Roleta podniesiona
			200 - Roleta opuszczona
<b>Zdarzenie</b>	<code>E.LCN.motor.S.M.R</code>	0	Roleta została zatrzymana
		1	Roleta w ruchu
<b>Komenda</b>	<code>C.LCN.motor.S.M.R</code>	up	Rusz w górę
		down	Rusz w dół
		stop	Zatrzymaj
		learn	Proces uczenia modułu LCN. Rolety wykonają pełne zamknięcie i otwarcie celem zmierzenia czasów ruchu rolety. Polecenie umożliwia późniejsze ustawianie rolety w konkretne położenie
		0-200	Ustawienie rolety na konkretne położenie
		forceUp	Wymuszony ruch rolety w górę.
		forceDown	Wymuszony ruch rolety w dół.
		limit;pos:xx	Ustawienie ograniczenia w ruchu rolety. xx stanowi maksymalną wartość do jakiej może przemieścić się roleta.
query	Odpytanie o bieżącą pozycję rolety.		



#### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach.

- **Ruch rolety w górę:**

**Nazwa:** `C.LCN.motor.0.10.1`

**Wartość:** up

Wykonaj ruch w górę rolety nr 1 w module o adresie 10.

- **Ustawienie określonej pozycji rolety:**

**Nazwa:** `C.LCN.motor.0.10.1`

**Wartość:** 100

Ustaw roletę nr 1 w module LCN nr 10 na połowę wysokości.

#### 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie reagujące na wykrycie ruchu rolety

**Kanał:** `E.LCN.motor.0.10.1`

**Dane:** 1

Roleta nr 1 w module z adresem 10 rozpoczęła ruch.



## LCN.motors

Ten identyfikator przeznaczony jest do sterowania jednocześnie maksymalnie 4 roletami podłączonymi do modułu LCN. Rekomendowane jest stosowanie tego identyfikatora do rolet z pozycjonowaniem. W przypadku rolet bez pozycjonowania należy używać identyfikatora LCN.relays.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.LCN.motors.S.M.R S - Segment M - Moduł R - Rolety: 1+2, 2+3, 3+4, 1-4 itd.	up	Rusz w górę
		down	Rusz w dół
		stop	Zatrzymaj
		learn	Proces uczenia modułu LCN. Rolety wykonają pełne zamknięcie i otwarcie celem zmierzenia czasów ruchu rolety. Polecenie umożliwia późniejsze ustawianie rolety w konkretne położenie
		0-200	Ustawienie rolety na konkretne położenie



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach.

- **Ruch rolety w górę:**

**Nazwa:** C.LCN.motor.0.10.1+2

**Wartość:** up

Wykonaj ruch w górę dla rolet nr 1 i 2 w module o adresie 10.

- **Ustawienie określonej pozycji rolety:**

**Nazwa:** C.LCN.motor.0.10.2+3

**Wartość:** 100

Ustaw rolety nr 2 i 3 w module LCN nr 10 na połowę wysokości.

- **Zamknięcie rolety**

**Nazwa:** C.LCN.motor.0.10.1-4

**Wartość:** down

Zamknij wszystkie cztery rolety.

### 2. Zdarzenia na zmianę stanu

Zdarzenie informujące o zmianie stanu będą w tym przypadku rozgłaszane dla pojedynczych rolet zgodnie z opisem identyfikatora LCN.motor.

## 12. Identyfikatory

### LCN.regulator

Identyfikator `LCN.regulator` służy do zarządzania regulatorami LCN. W przypadku regulacji temperatury, należy pamiętać, że w systemie LCN temperatura jest wyrażana w innej skali. Przeliczenie ze stopni Celsjusza na skalę LCN wykonywane jest wg następującej formuły:  $temperatura * 10 + 1000$ , zatem:

- $21.5^{\circ}\text{C}$  będzie zapisane jako 1215
- $-11^{\circ}\text{C}$  będzie zapisane jako 990

Moduły z firmware starszym niż 17xxxx dysponują zakresem nastaw 0-65534, natomiast moduły nowsze zakresem od 0 do 32767.

Zasada działania regulatora LCN umożliwia ustawienie wartości absolutnej w regulatorze jedynie w zakresie od 0 do 3047. Aby ustawiać wartości w pełnym zakresie pracy regulatora (0-65534 lub 0-32767 w zależności od wersji firmware) należy wysłać komendy zmieniające wartość względem wartości aktualnej (komenda `current:change`).

	Identyfikator	Wartość	Opis
<b>Stan</b>	<code>LCN.regulator.S.M.R</code> S - Segment M - Moduł R - Regulator (1,2,t)	0-65534 lub 0-32767	Bieżąca wartość regulatora
<b>Zdarzenie</b>	<code>E.LCN.regulator.S.M.R</code>	0-65534 lub 0-32767	Ustawiona została wartość z zakresu pracy regulatora
<b>Komenda</b>	<code>C.LCN.regulator.S.M.R</code>	lock	Zablokuj regulator
		unlock	Odblokuj regulator
		<code>current;change:ZM</code>	Zmień o wartość <b>ZM</b> (+/-) względem aktualnej wartości regulatora. Zakres zmian: -1023 do 1023.
		<code>programmed;change:ZM</code>	Zmień o wartość <b>ZM</b> (+/-) względem zapamiętanej wartości regulatora. Zakres zmian: -1023 do 1023.
		0-3047	Ustawienie regulatora



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach.

- **Ustawienie wartości zadanej (absolutnej):**

**Nazwa:** C.LCN.regulator.0.10.1

**Wartość:** 1250

Ustaw wartość zadaną regulatora nr 1 na 1250, moduł LCN nr 10. W przypadku sterowania temperaturą oznacza to ustawienie temperatury zadanej na 25°C.

- **Blokada regulatora:**

**Nazwa:** C.LCN.regulator.0.10.1

**Wartość:** lock

Zablokuj regulator nr 1 w module o adresie 10.

- **Relatywna zmiana wartości zadanej.**

**Nazwa:** C.LCN.regulator.0.10.1

**Wartość:** current;change:-20

Ustaw jako zadaną wartość regulatora aktualną wartość pomniejszoną o 2°C.

### 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie reagujące na zmianę wartości zadanej regulatora.

**Kanał:** E.LCN.regulator.0.10.1

**Dane:** 1225

Wartość zadana regulatora nr 1 w module LCN o adresie 10 została ustawiona na 1225, czyli 22,5°C.

## 12. Identyfikatory

### LCN.value

Identyfikator `LCN.value` stanowi odwołanie do odczytu temperatur z czujników LCN oraz umożliwia emitowanie zdarzeń o zmianach temperatury mierzonej.

Wartości zwracane przez ten identyfikator są wyrażone w skali LCN, opisanej w przypadku identyfikatora `LCN.regulator`.

W przypadku modułów LCN, które posiadają 12 zmiennych, identyfikator `LCN.value` jest wciąż aktualny, kolejno `t`, `r1` oraz `r2` odpowiadają zmiennym od 1 do 3.

	Identyfikator	Wartość	Opis
Stan	<code>LCN.value.S.M.C</code> s - Segment m - Moduł c - Czujnik (r1, r2, t - R1Var, R2Var, TVar).	0-2000	Wyświetlenie temperatury bieżącej.
	<code>LCN.value.S.M.s0x</code> x - nr licznika	liczba	Wartość impulsów zliczonych przez licznik S0 w module LCN.
Zdarzenie	<code>E.LCN.value.S.M.C</code>	0-2000	Zdarzenie o zmianie mierzonej temperatury.
Komenda	<code>C.LCN.value.S.M.t</code> Dopuszczalna jest zmiana tylko zmiennej Tvar	change:xx	Relatywna zmiana wartości zmiennej. Dopuszczalny zakres zmian +/- 4000. Komenda przydatna szczególnie, gdy potrzebne jest skompensowanie mierzonej temperatury.
		reset	Wyzerowanie zmian wartości.



#### 1. Komenda na zmianę wartości zmiennej

- **Kanał:** `C.LCN.value.0.10.t`  
**Dane:** `change:100`  
Zmiana aktualnej wartości zmiennej TVar o 100, moduł nr 10.
- **Kanał:** `C.LCN.value.0.10.t`  
**Dane:** `reset`  
Zresetowanie zmian w wartości zmiennej Tvar, moduł nr 10.

#### 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie reagujące na zmianę wartości zmiennej nr R1Var.  
**Kanał:** `E.LCN.value.0.10.r1`  
**Dane:** `1250`  
Temperatura mierzona przez czujnik przypisany do zmiennej R1Var modułu LCN o adresie 10 osiągnęła wartość 25°C.

## LCN.variable

Nowsza generacja modułów LCN udostępnia 12 zmiennych (wcześniej 3). Identyfikator `LCN.variable` pozwala odczytywać wartości tych zmiennych oraz informuje przez zdarzenia o zmianie tychże wartości. Dostępne wcześniej zmienne `R1Var`, `R2Var` oraz `TVar` można wciąż odczytać przy użyciu identyfikatorów `LCN.value`, zgodnie z opisem z poprzedniego podrozdziału, a także przy użyciu identyfikatora `LCN.variable` kolejno pod numerami zmiennych 1,2 i 3.

	Identyfikator	Wartość	Opis
<b>Stan</b>	<code>LCN.variable.S.M.Z</code> s - Segment m - Moduł z - Zmienna (od 1 do 12)	0-2000	Wyświetlenie bieżącej wartości danej zmiennej.
<b>Zdarzenie</b>	<code>E.LCN.variable.S.M.Z</code>	0-2000	Zdarzenie informujące o zmianie wartości danej zmiennej.
<b>Komenda</b>	<code>C.LCN.variable.S.M.Z</code>	<code>change:xx</code>	Relatywna zmiana wartości zmiennej. Dopuszczalny zakres zmian +/- 4000. Komenda przydatna szczególnie, gdy potrzebne jest skompensowanie mierzonej temperatury.
		<code>reset</code>	Wyzerowanie zmian wartości.



### 1. Komenda na zmianę wartości zmiennej

- **Kanał:** `C.LCN.variable.0.10.1`  
**Dane:** `change:100`  
Zmiana aktualnej wartości zmiennej nr 1 o 100, moduł nr 10.
- **Kanał:** `C.LCN.variable.0.10.1`  
**Dane:** `reset`  
Zresetowanie zmian w wartości zmiennej nr 1, moduł nr 10.

### 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie reagujące na zmianę wartości zmiennej nr 1.  
**Kanał:** `E.LCN.variable.0.10.1`  
**Dane:** `100`  
Zmienna nr 1 w module LCN o adresie 10 osiągnęła wartość 100.

## 12. Identyfikatory

### LCN.threshold (moduły z firmware starszym niż 17xxxx)

Identyfikator `LCN.threshold` służy do sterowania wartościami progowymi modułów LCN oraz informowania o zmianach tychże wartości i wyświetlania ich stanu. Zmiany wartości progowej zawsze wykonywane są względem wartości bieżącej lub zaprogramowanej w module LCN. Sposób kodowania wartości jest analogiczny, jak w przypadku regulatorów. Zatem zmiana wartości o 10 powoduje zmianę w 0.1 w module LCN!

	Identyfikator	Wartość	Opis
<b>Stan</b>	<code>LCN.threshold.S.M.W</code> s - Segment m - Moduł w - Wartość progowa (1-5)	0-32767	Bieżący stan wartości progowej
<b>Zdarzenie</b>	<code>E.LCN.threshold.S.M.W</code>	0-32767	Ustawiona została wartość z zakresu od 0 do 32767
<b>Komenda</b>	<code>C.LCN.threshold.S.M.W</code>	<code>current;change:ZM</code>	Zmień o wartość <b>ZM</b> (+/-) względem aktualnej wartości. Zakres zmian: -1023 do 1023
		<code>programmed;change:ZM</code>	Zmień o wartość <b>ZM</b> (+/-) względem zapamiętanej wartości. Zakres zmian: -1023 do 1023.
		-1023 do 1023	Zmiana wartości progowej o zadaną liczbę. Zmiany są kodowane identycznie, jak temperatura w przypadku regulatorów. Jest to alternatywna komenda do komendy: <code>current;change:ZM</code>



#### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach.

- **Ustawienie wartości zadanej:**

**Nazwa:** `C.LCN.threshold.0.10.1`

**Wartość:** -100

Zmniejsz bieżącą wartość progową nr 1 o 100, w module LCN o adresie 10.

**Nazwa:** `C.LCN.threshold.0.10.2`

**Wartość:** `programmed;change:20`

Zwiększ wartość progową nr 2 względem wartości zaprogramowanej o 20, moduł o adresie 10.

#### 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie reagujące na zmianę wartości progowej

**Kanał:** `E.LCN.threshold.0.10.1`

**Dane:** 1225

Wartość progowa nr 1 w module LCN o adresie 10 została ustawiona na 1225, czyli 22,5°C.

## LCN.threshold (moduły z firmware 17xxxx i nowszym)

Identyfikator `LCN.threshold` służy do sterowania wartościami progowymi modułów LCN oraz informowania o zmianach tychże wartości i wyświetlania ich stanu. Zmiany wartości progowej zawsze wykonywane są względem wartości bieżącej lub zaprogramowanej w module LCN. Sposób kodowania wartości jest analogiczny, jak w przypadku regulatorów. Zatem zmiana wartości o 10 powoduje zmianę w 0,1 w module LCN!

Moduły nowszej generacji mają 4x więcej wartości progowych, dzięki wprowadzeniu rejestrów wartości progowych.

	Identyfikator	Wartość	Opis
<b>Stan</b>	<code>LCN.threshold.S.M.R.W</code> S - Segment M - Moduł R - Rejestr (1-4) W - Wartość progowa (1-4)	0-32767	Bieżący stan wartości progowej
<b>Zdarzenie</b>	<code>E.LCN.threshold.S.M.R.W</code>	0-32767	Ustawiona została wartość z zakresu od 0 do 32767
<b>Komenda</b>	<code>C.LCN.threshold.S.M.R.W</code>	<code>current; change: ZM</code>	Zmień o wartość <b>ZM</b> (+/-) względem aktualnej wartości. Zakres zmian: -1023 do 1023.
		<code>programmed; change: ZM</code>	Zmień o wartość <b>ZM</b> (+/-) względem zapamiętanej wartości. Zakres zmian: -1023 do 1023.
		-1023 do 1023	Zmiana wartości progowej o zadaną liczbę. Zmiany są kodowane identycznie, jak temperatura w przypadku regulatorów. Jest to alternatywna komenda do komendy: <code>current; change: ZM</code>



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach.

- **Ustawienie wartości zadanej:**

**Nazwa:** `C.LCN.threshold.0.10.1.1`

**Wartość:** -100

Zmniejsz bieżącą wartość progową nr 1, w rejestrze nr 1, o 100, moduł LCN o adresie 10.

**Nazwa:** `C.LCN.threshold.0.10.1.2`

**Wartość:** `programmed; change: 20`

Zwiększ wartość progową nr 2, w rejestrze 1 względem wartości zaprogramowanej o 20, moduł o adresie 10.

### 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie reagujące na zmianę wartości progowej

**Kanał:** `E.LCN.threshold.0.10.1.1`

**Dane:** 1225

Wartość progowa nr 1 w rejestrze 1, w module LCN o adresie 10 została ustawiona na 1225, czyli 22,5°C.

## 12. Identyfikatory

### LCN.key

Identyfikator LCN . key informuje o odebraniu polecenia LCN „wyślij klawisz” wysłanego na adres modułu DOMIQ/ Base.

W systemie LCN nie ma możliwości pasywnego odczytu przyciśniętych klawiszy, należy zawsze wykorzystać funkcję „wyślij klawisz” lub „ładuj scenę” aby przypisać akcje do przycisków.

	Identyfikator	Wartość	Opis
<b>Zdarzenie</b>	E . LCN . key . <b>S . M . TP</b> S - Segment M - Moduł T - Tablica: (A,B,C) P - Przycisk: (1-8)	hit/make/break	Odebrano pakiet „wyślij klawisze”



#### 1. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- **Kanał:** E . LCN . key . 0 . 10 . A1

**Dane:** make

Odebrano polecenie „wyślij klawisz”, tablica A, klawisz 1, komenda długo (make). Polecenie zostało wysłane z modułu o adresie 10.

- **Kanał:** E . LCN . key . 0 . 36 . B5

**Dane:** hit

Odebrano polecenie „wyślij klawisz”, tablica B, klawisz 5, komenda krótko (hit). Polecenie zostało wysłane z modułu o adresie 36.



## LCN.sendkey

Komenda `LCN.sendkey` wykonuje polecenie „wyślij klawisz”. Komenda może być wysłana do dowolnego modułu LCN, na jedną z czterech tablic, z każdą z dostępnych akcji (krótko/długo/puść). Po odebraniu tego polecenia przez moduł LCN, wykonana zostanie akcja przypisana do danego klawisza w oprogramowaniu LCN-Pro. Dzięki tej komendzie można utworzyć przyciski np. w aplikacji Remote lub na wizualizacji, które będą wykonywały akcje zaprogramowane bezpośrednio w modułach LCN.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<code>C.LCN.sendkey.S.M.TP</code> <b>S</b> - Segment <b>M</b> - Moduł <b>T</b> - Tablica: (A,B,C,D) <b>P</b> - Przycisk: (1-8)	hit/make/break	Komenda wyślij klawisz.



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- Nazwa:** `C.LCN.sendkey.0.10.A1`  
**Wartość:** hit  
 Wyślij polecenie hit (krótko) na klawisz 1 w tablicy A, w module o adresie 10, segment 0.
- Nazwa:** `C.LCN.sendkey.0.10.B3`  
**Wartość:** make  
 Wyślij polecenie make(długo) na klawisz 3 w tablicy B, w module o adresie 10, segment 0.

## LCN.scene

Zdarzenie o identyfikatorze `LCN.scene` informuje o odebraniu pakietu „ładuj scenę” wysłanego na adres modułu DOMIQ/Base.

	Identyfikator	Wartość	Opis
<b>Zdarzenie</b>	<code>E.LCN.scene</code>	1-100	Numer wywołanej sceny w zakresie od 1 do 100.



### 1. Zdarzenia informujące o załadowaniu sceny

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Kanał:** `E.LCN.scene`  
**Dane:** 12  
 Załadowana została scena nr 12

## 12. Identyfikatory

### LCN.scenes

Komenda `LCN.scenes` służy do wczytywania/zapisywania scen LCN. Umożliwia odczyt i zapisywanie scen zarówno na wyjściach ściemnianych i przekaźnikowych. Sposób działania modułów LCN uniemożliwia jednak jednoczesne ładowanie/zapis scen kombinowanych (zawierających oba rodzaje wyjść). Aby załadować/zapisać scenę kombinowaną należy wysłać dwie niezależne komendy. Domyślnie zapisywane/odczytywane są sceny wykorzystujące wyjścia ściemniane. Jeżeli w składni polecenia zostanie użyty numer rejestru, a następnie wysłana zostanie komenda bez wskazania konkretnego rejestru, wówczas komenda zostanie wysłana na rejestr wskazany w komendzie poprzedniej. Do momentu wysłania komendy ze wskazaniem rejestru, wszystkie komendy wysyłane są domyślnie do rejestru nr 0.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<code>C.LCN.scenes.S.M</code> S - Numer segmentu M - Numer modułu	<b>A:N;Reg;T:O;R</b> A - Akcja (load/save) N - Numer sceny (od 1 do 10) Reg (opcjonalnie) - Numer rejestru T - Typ wyjścia (outputs, relays) O - Numery wyjść R (opcjonalnie) - Parametr <b>ramp</b>	Zapis lub załadowanie wybranej sceny.



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

Wszystkie przykłady odnoszą się do modułu o adresie 20, segment nr 0.

- **Nazwa:** C.LCN.scenes.0.20  
**Wartość:** load:1  
Załaduj scenę nr 1, wszystkie wyjścia ściemniane.
- **Nazwa:** C.LCN.scenes.0.20  
**Wartość:** load:3;outputs:010;ramp:10  
Załaduj scenę nr 3, użyj wyjścia ściemnianego nr 2, ustaw rampę równą 10.
- **Nazwa:** C.LCN.scenes.0.20  
**Wartość:** save:1;outputs:110  
Zapisz scenę nr 1, użyj wyjść ściemnianych nr 1 i 2.
- **Nazwa:** C.LCN.scenes.0.20  
**Wartość:** save:1;ramp:2  
Zapisz scenę nr 1, wszystkie wyjścia ściemniane.
- **Nazwa:** C.LCN.scenes.0.20  
**Wartość:** load:1;relays:11-----  
Załaduj pierwszą scenę, użyj przekaźników nr 1 i 2.
- **Nazwa:** C.LCN.scenes.0.20  
**Wartość:** load:2;relays:11111111  
Załaduj drugą scenę, użyj wszystkich przekaźników.
- **Nazwa:** C.LCN.scenes.0.20  
**Wartość:** save:2;relays:11-----  
Zapisz scenę nr 2, użyj przekaźników nr 1 i 2.
- **Nazwa:** C.LCN.scenes.0.20  
**Wartość:** load:1;register:5;outputs:1100;ramp:3;  
Załaduj scenę nr 1 w rejestrze nr 5, użyj wyjść ściemnianych nr 1 i 2 z rampą równą 3.

## 12. Identyfikatory

### LCN.transponder

Identyfikator `LCN.transponder` służy do rozgłaszania zdarzeń o otrzymaniu kodu transpondera LCN.

Wykorzystując te zdarzenia, możliwa jest implementacja systemów kontroli dostępu, rejestracji ruchu obiektów etc.

	Identyfikator	Wartość	Opis
<b>Zdarzenie</b>	<code>E.LCN.transponder.S.M</code> S - segment M - moduł	<code>KKKKKK P A</code> KKKKKK - kod transpondera P - Numer przycisku A - Akcja (hit, make, break lub other)	Moduł LCN odczytał kod transpondera (IR lub RFID)

Typ akcji `other` jest generowany w przypadku użycia transpondera innego producenta niż LCN.



#### 1. Zdarzenia informujące o odebraniu kodu transpondera

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- **Zdarzenie na odebranie określonego kodu i określonej akcji**

**Kanał:** `E.LCN.transponder.0.10`

**Dane:** `B308FF 1 hit`

Odebrano kod transpondera `B308FF`. Zasympulowano wciśnięcie przycisku nr 1 w module LCN o adresie 10.

- **Zdarzenie reagujące na odebranie dowolnego kodu transpondera z określonego modułu LCN.**

**Kanał:** `E.LCN.transponder.0.10`

**Dane:** pozostaw to pole puste

Wartość z jaką zostało wywołane zdarzenie dostępna jest pod parametrem `$D0`. Można ją przekazać do np. do skryptu zarządzającego kontrolą dostępu.

## LCN.dali

Komendy z grupy `LCN.dali` służą do sterowania balastami DALI podłączonymi do wyjść modułów LCN. Implementacja standardu DALI w LCN umożliwia jedynie wysyłanie komend do balastów DALI, jednakże nie są otrzymywane żadne informacje zwrotne z balastów. Komendy można podzielić ze względu na obszar zastosowania. Pierwszą podgrupą są komendy wysyłane do indywidualnych balastów DALI. Drugą podgrupę stanowią komendy, które są przeznaczone do sterowania jednocześnie całą instalacją DALI. Trzecią grupą są identyfikatory grupowe, które są wykorzystywane do sterowania grupami oświetlenia DALI.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<b>C.LCN.dali.S.M.T.N</b> <b>S</b> - Numer segmentu <b>M</b> - Numer modułu <b>T</b> - Typ komendy. Dostępne są trzy typy komend: <b>evg</b> - komenda wysyłana do pojedynczego balastu DALI. <b>group</b> - komenda sterująca grupą oświetlenia DALI. <b>all</b> - komenda sterująca całą instalacją. <b>N</b> - (tylko dla typów komend <b>evg</b> i <b>group</b> ) - Adres balastu/numer grupy	0-254	Ustaw jasność.
		on	Załączenie wybranej lampy. Ustawiona zostanie jasność wg parametru <b>Poziom max</b> .
		off	Wyłączenie wybranej lampy.
		up	Skokowe rozjaśnianie.
		down	Skokowe ściemnianie.
		stepup	Skokowa rozjaśnianie o 1. Komenda nie powoduje załączenia lampy.
		stepdown	Skokowe ściemnianie o 1. Komenda nie powoduje wyłączenia lampy.
		load:xx	Załadowanie wybranej sceny, gdzie <b>xx</b> oznacza numer sceny.
		save:xx	Zapisanie bieżącej jasności do sceny, gdzie <b>xx</b> oznacza numer sceny.
		dtr:xx	Zapisanie wartości w rejestrze dtr. „ <b>xx</b> ” oznacza zapisywaną wartość.



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- Nazwa:** `C.LCN.dali.0.10.evg.1`  
**Wartość:** 100  
 Ustaw wartość 100 (max) w balaście o adresie 1 podłączonym do modułu LCN o adresie 10.
- Nazwa:** `C.LCN.dali.0.10.all`  
**Wartość:** off  
 Wyłącz wszystkie balasty w instalacji podłączonej do modułu LCN o adresie 10.
- Nazwa:** `C.LCN.dali.0.10.group.5`  
**Wartość:** load:3  
 Załaduj scenę nr 3 w balastach DALI przynależnych do grupy nr 5.

## 12. Identyfikatory

### LCN.locks

Komenda blokująca klawisze w dowolnej tabeli modułu LCN.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.LCN.locks.S.M.T S - Segment M - Moduł T - Tabela (A,B,C,D)	0	Nie blokuj
		1	Zablokuj
		T	Przełącz
		-	Nie zmieniaj stanu blokady



#### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Nazwa:** C.LCN.locks.0.10.A

**Wartość:** 0011----

Odblokuj klawisze nr 1 i 2, zablokuj klawisze nr 3 i 4 w tabeli A, w module o adresie 10. Pozostałe klawisze w tabeli pozostaw bez zmian.

### LCN.text

Komenda LCN.text umożliwia wyświetlanie dowolnych tekstów na wyświetlaczach przycisków ściennych LCN z serii GT-D.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.LCN.text.S.M.L S - Segment M - Moduł T - Linia tekstu (1-4)	dowolny tekst	Wyświetlenie dowolnego tekstu, w wybranej linii, na wyświetlaczu panela LCN-GT-D.



#### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Nazwa:** C.LCN.text.0.10.1

**Wartość:** Tekst do wyświetlenia

Wyświetl napis „Tekst do wyświetlenia” w pierwszej linii wyświetlacza przycisków podłączonych do modułu LCN o adresie 10.

## LCN.groups

Komenda `LCN.groups` umożliwia zarządzanie grupami LCN.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<code>C.LCN.groups.S.M</code> S - Segment M - Moduł T - Linia tekstu (1-4)	<code>clear</code>	Komenda kasująca jednocześnie przypisanie do wszystkich grup.
		<code>join:xx</code>	Komenda przypisująca moduł LCN do wybranej grupy. <b>xx</b> oznacza numer grupy.
		<code>leave:xx</code>	Komenda kasująca przypisanie do wybranej grupy. <b>xx</b> oznacza numer grupy.



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Usunięcie wszystkich przypisań do grup**

**Nazwa:** `C.LCN.groups.0.10`

**Wartość:** `clear`

Skasuj przypisanie do wszystkich grup modułu 10.

- **Przypisanie modułu do wybranej grupy**

**Nazwa:** `C.LCN.groups.0.10`

**Wartość:** `join:5`

Przypisz moduł LCN o adresie 10 do grupy nr 5.

- **Usunięcie przypisania z wybranej grupy**

**Nazwa:** `C.LCN.groups.0.10`

**Wartość:** `leave:6`

Usuń przypisanie modułu o adresie 10 z grupy nr 8.

## 12.3. IDS

Identyfikatory z grupy **IDS** są dedykowane do integracji urządzeń DOMIQ z centralami alarmowymi Satel Integra. Poniższa tabela zawiera przegląd wszystkich identyfikatorów **IDS** i ich główne cechy. Identyfikatory dostępne wyłącznie w przypadku integracji poprzez moduły Satel ETHM-1 zostały opisane jako "tylko ETHM". Szczegółowy opis poszczególnych identyfikatorów wraz z przykładami zastosowania znajduje się w dalszej części podrozdziału.

Identyfikator	Skrócony opis	Stan	Zdarzenia	Komenda
IDS.input	Stan i informacje o zmianach stanu pojedynczego wejścia centrali alarmowej.	+	+	-
IDS.output	Sterowanie i stan pojedynczego wyjścia centrali alarmowej.	+	+	+
IDS.armed	Sterowanie i stan pojedynczej strefy alarmowej.	+	+	+
IDS.entry	Zdarzenie informujących o naruszeniu strefy z uruchomionym odliczaniem czasu na wejście.	-	+	-
IDS.exit	Zdarzenie informujących o naruszeniu strefy z uruchomionym odliczaniem czasu na wejście.	-	+	-
IDS.alarm	Zdarzenie informujące o alarmie w danej strefie.	-	+	-
IDS.log	Pojedynczy wpis z dziennika zdarzeń centrali alarmowej.	+	-	-
IDS.log.new	Zbiorczy log zdarzeń z centrali alarmowej wygenerowany dla bieżącego dnia.	+	+	-
IDS.problem	Stan i informacje o zmianach stanu dotyczącego awarii w centrali alarmowej.	+	+	-
IDS.clearlog	Komenda kasująca dane logów pobranych z centrali alarmowej	-	-	+
IDS.scan	Komenda inicjująca pełen odczyt danych z centrali alarmowej	-	-	+



## IDS.input

Identyfikator `IDS.input` przechowuje stan pojedynczego wejścia centrali alarmowej oraz emituje zdarzenie o zmianach stanu.

	Identyfikator	Wartość	Opis
<b>Stan</b>	<code>IDS.input.N</code> N - numer wejścia	0	Wejście nieaktywne
		1	Wejście aktywne
<b>Zdarzenie</b>	<code>E.IDS.input.N</code>	0	Wejście stało się nieaktywne
		1	Wejście stało się aktywne



### 1. Zdarzenia informujące o zmianie stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- **Kanał:** `E.IDS.input.1`

**Dane:** 1

Wejście alarmowe nr 1 zmieniło wartość na 1. Wejściem alarmowym może być np. czujka PIR, kontaktron itd. Jeżeli założymy, że naszym wejściem alarmowym jest kontaktron zamontowany na ramie okiennej, wówczas powyższy przykład należy rozumieć jako: Otwarto okno nr ... .

## IDS.output

Identyfikator `IDS.output` służy do zarządzania pojedynczym wyjściem centrali alarmowej.

Każda komenda mająca zmienić wartość wyjścia centrali alarmowej Satel wymaga podania kodu PIN centrali.

	Identyfikator	Wartość	Opis
<b>Stan</b>	<code>IDS.output.N</code> N – Numer wyjścia	0	Wyjście nieaktywne.
		1	Wyjście aktywne.
<b>Zdarzenie</b>	<code>E.IDS.output.N</code>	0/off	Wyjście stało się nieaktywne.
		1/on	Wyjście stało się aktywne.
<b>Komenda</b>	<code>C.IDS.output.N</code>	0/off;pin:xxxx	Wyłącz wyjście.
		1/on;pin:xxxx	Włącz wyjście.
		toggle;pin:xxxx	Zmień stan wyjścia na przeciwny.



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Aktywowanie wyjścia**

**Nazwa:** C.IDS.output.10

**Wartość:** 1;pin:12345

Włącz wyjście nr 10.

- **Deaktywacja wyjścia**

**Nazwa:** C.IDS.output.10

**Wartość:** off;pin:12345

Wyłącz wyjście nr 10.

- **Przełączenie stanu na przeciwny:**

**Nazwa:** C.IDS.output.10

**Wartość:** toggle;pin:12345

Przełącz wyjście nr 10 na stan przeciwny.

### 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie reagujące na załączenie wyjścia.

**Kanał:** E.IDS.output.10

**Dane:** 1

Wyjście nr 10 zostało włączone.

- Zdarzenie wyzwalane dowolną zmianą wartości wyjścia:

**Kanał:** E.IDS.output.10

**Dane:** pozostaw to pole puste

Wartość z jaką zostało wywołane zdarzenie dostępna jest pod parametrem \$D0.

## IDS.armed

Identyfikator `IDS.armed` służy zarządzania pojedynczą strefą alarmową.

Każda komenda zmieniająca stan strefy alarmowej wymaga potwierdzenia kodem PIN.

	Identyfikator	Wartość	Opis
<b>Stan</b>	<code>IDS.armed.s</code>	0	Strefa rozbrojona
	<code>s</code> - Numer strefy	1	Strefa uzbrojona
<b>Zdarzenie</b>	<code>E.IDS.armed.s</code>	0	Strefa została rozbrojona
		1	Strefa została uzbrojona
<b>Komenda</b>	<code>C.IDS.armed.s</code>	<code>0/off;pin:xxxx</code>	Rozbrój strefę
		<code>1/on/arm;pin:xxxx</code>	Uzbrój strefę



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Uzbrojenie strefy**

**Nazwa:** `C.IDS.armed.1`

**Wartość:** `1;pin:12345`

Uzbrój strefę nr 1.

- **Rozbrojenie strefy**

**Nazwa:** `C.IDS.armed.1`

**Wartość:** `0;pin:12345`

Rozbrój strefę nr 1.

### 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie wyzwalane przez uzbrojenie strefy. Ten rodzaj zdarzenia może posłużyć jako inicjator działań w systemie po opuszczeniu domu przez mieszkańców np. obniżenie temperatur, zamknięcie zaworu wody itd.

**Kanał:** `E.IDS.armed.1`

**Dane:** 1

Strefa nr 1 została uzbrojona.

- Zdarzenie wyzwalane dowolną zmianą wartości wyjścia:

**Kanał:** `E.IDS.output.10`

**Dane:** pozostaw to pole puste

Wartość z jaką zostało wywołane zdarzenie dostępna jest pod parametrem `$D0`.

## 12. Identyfikatory

### IDS.entry

Identyfikator `IDS.entry` wykorzystywany jest do informowania o odliczaniu czasu po naruszeniu strefy z określonym czasem na wejście. Czas na wejście w strefie alarmowej oznacza czas, po którym zostanie uruchomiony alarm, jeżeli użytkownik go nie rozbroi.

Jeżeli w centrali alarmowej zostanie zaprogramowana strefa z określonym czasem na wejście można zdefiniować akcję, która zostanie zrealizowana w przypadku pojawienia się człowieka w tej strefie np. zapalenie światła w przedpokoju itp.

	Identyfikator	Wartość	Opis
Zdarzenie	E.IDS.entry.s s – numer strefy	0	Odliczanie zakończone, po chwili pojawi się zdarzenie o rozbrojeniu czuwania lub o alarmie
		1	Rozpoczęto odliczanie czasu na wejście



#### 1. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie na wykrycie ruchu w strefie z czasem na wejście

**Kanał:** E.IDS.entry.1

**Dane:** 1

Naruszenie strefy alarmowej nr 1, która ma ustawiony czas na wejście.

### IDS.exit

Identyfikator `IDS.exit` informuje o rozpoczęciu odliczania czasu w strefie alarmowej z określonym czasem na wyjście. W tym czasie ignorowane są niektóre wejścia alarmowe, aby możliwe było opuszczenie strefy. Przez czas na wyjście należy rozumieć czas, po którym zostanie uruchomiony alarm, jeżeli użytkownik nie opuści strefy po jej uzbrojeniu. Jeżeli w centrali alarmowej zostanie zaprogramowana strefa z określonym czasem na wyjście, wówczas w systemie DOMIQ można zdefiniować akcję, która zostanie zrealizowana po opuszczeniu strefy alarmowej przez człowieka, np. zgaszenie wszystkich światel w budynku.

	Identyfikator	Wartość	Opis
Zdarzenie	E.IDS.exit.s s – numer strefy	0	Zakończono odliczanie, po chwili pojawi się zdarzenie o uzbrojeniu strefy
		1	Rozpoczęto odliczanie czasu na wyjście



#### 1. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie na wykrycie ruchu w strefie z czasem na wyjście

**Kanał:** E.IDS.exit.1

**Dane:** 1

Naruszenie strefy alarmowej nr 1, która ma ustawiony czas na wyjście.

## IDS.alarm

Zadaniem identyfikatora `IDS.alarm` jest informowanie użytkownika o alarmie w strefie.

	Identyfikator	Wartość	Opis
Zdarzenie	<code>E.IDS.alarm.s</code>	0	Alarm wyłączony
	<code>s</code> - Numer strefy alarmowej	1	Wzbudzenie alarmu.



### 1. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie w wypadku alarmu w strefie

**Kanał:** `E.IDS.alarm.1`

**Dane:** 1

Alarm w strefie 1.

Bardzo uogólniony przypadek:

**Kanał:** `E.IDS.alarm. (%d+)`

**Dane:** 1

Zdarzenie zostanie wywołane przy wykryciu alarmu w dowolnej strefie, a nr strefy alarmowej jest wówczas dostępny przy użyciu parametru `§C1`.

## 12. Identyfikatory

### IDS.log (tylko ETHM)

Ten identyfikator przechowuje informację o pojedynczym zdarzeniu odczytanym z dziennika zdarzeń centrali alarmowej. System DOMIQ może pobierać dane o zdarzeniach z centrali w zakresie od 1 do 7 dni, w zależności od przyjętej konfiguracji.

	Identyfikator	Wartość
<b>Stan</b>	IDS.log.DD.MM.RRRR.#NUM DD - Dzień MM - Miesiąc RRRR - Rok #NUM - Numer zdarzenia danego dnia	dane o zdarzeniu zapisane w formacie: godzina   opis   strefa/urządzenie   użytkownik  Ostatni parametr nie jest obecny we wszystkich rodzajach zdarzeń.



#### 1. Odczyt pojedynczego zdarzenia z centrali alarmowej

- IDS.log.29.01.2025.#1 = 12:55 | Skasowan. alarmu | Biuro | Serwis  
Zdarzenie nr 1 w dniu 29.01.2025. Zdarzenie zarejestrowane o 12:55 w strefie o nazwie Biuro. Akcja skasowania alarmu wykonana przez użytkownika o nazwie Serwis.
- IDS.log.29.01.2025.#12 = 16:12 | Kabel LAN ok | ETHM-1 (3)  
Zdarzenie nr 12 w dniu 29.01.2025. Zdarzenie zarejestrowane o 16:12 informujące o prawidłowym podłączeniu kabla LAN do modułu ETHM-1 (adres 3).

## IDS.log.new (tylko ETHM)

Ten identyfikator przechowuje dane zbiorczego logu zdarzeń z centrali alarmowej dla bieżącego dnia. Maksymalna pojemność logu to 20 najnowszych zdarzeń. Starsze zdarzenia są wciąż dostępne z użyciem identyfikatora `IDS.log`. Z użyciem identyfikatora `IDS.log.new` z łatwością wyświetlisz najświeższe logi o zdarzeniach w aplikacji Remote.

	Identyfikator	Wartość
<b>Stan</b>	<code>IDS.log.new</code>	dane o zdarzeniach (każde zdarzenie w osobnej linii)
<b>Zdarzenie</b>	<code>E.IDS.log.new</code>	dane o zdarzeniach (każde zdarzenie w osobnej linii)



### 1. Wyświetlenie logu w aplikacji Remote

- W zakładce **Remote** dodaj kontrolkę **Tekst** i otwórz ją w trybie edycji (kliknij na niej 2x).
- **Kanał:** `IDS.log.new`
- Zapisz zakładkę **Remote**.

### 2. Zdarzenie dla nowych logów

Jeżeli w Twojej instalacji zachodzi potrzeba monitorowania nowych zdarzeń w centrali alarmowej, system DOMIQ może przechwycić je generując zdarzenie. Poniżej przykładowa definicja zdarzenia:

- **Kanał:** `E.IDS.log.new`
- **Dane:** pozostaw puste pole - zdarzenie wykona się wówczas dla każdej zmiany zawartości logu.
- W sekcji **Akcje** zdefiniuj jakie czynności mają być wykonane w reakcji na zdarzenie.

## 12. Identyfikatory

### IDS.problem (tylko ETHM)

Identyfikator przechowuje informacje o awariach zarejestrowanych w systemie alarmowym. W obecnej implementacji pobierane są dane o awariach z grupy 4, czyli awarie ekspanderów i manipulatorów. Poniższa tabela zawiera opis obsługiwanych typów awarii wraz wartością parametru P, który jest używany w identyfikatorze IDS.problem.

Typ awarii (P)	Opis
expandernocom	brak komunikacji z ekspanderem
expsubstituted	podmiana ekspandera
keypadnocom	brak komunikacji z manipulatorem
keypadsubstituted	podmiana manipulatora
cableproblem	brak kabla w module ETHM-1
expandertamper	sabotaż ekspandera
keypadtamper	sabotaż manipulatora
keypadiniterr	błąd inicjalizacji manipulatora

	Identyfikator	Wartość	Opis
<b>Stan</b>	IDS.problem.P.A P - typ awarii A - adres urządzenia	0/1	Status dla bieżącego typu awarii. 1 oznacza, że awaria występuje.
<b>Zdarzenie</b>	E.IDS.problem.P.A	0	Dana awaria przestała występować.
		1	Dana awaria zaczęła występować.



#### 1. Wyświetlenie statusu o awarii w aplikacji Remote

Przykład prezentuje wyświetlenie informacji o sabotażu manipulatora o adresie 1.

- W zakładce **Remote** dodaj kontrolkę **Status** i otwórz ją w trybie edycji.
- **Etykieta:** Wpisz dowolny opis kontrolki.
- **Kanał:** *IDS.problem.keypadtamper.1*
- Opcje **Ikona wł.** oraz **Ikona wył.** ustal wg własnych preferencji.

#### 2. Zdarzenie na wystąpieniu awarii

Przykład prezentuje zdarzenie na zanik komunikacji z ekspanderem o adresie 5.

- Dodaj nowe zdarzenie.
- **Kanał:** *E.IDS.problem.expandernocom.5*
- **Dane:** *1*
- W sekcji **Akcje** zdefiniuj akcje, które mają zostać wykonane w związku z tym zdarzeniem.



## IDS.clearlog (tylko ETHM)

Komenda służąca do kasowania logu o zdarzeniach pobranych z centrali alarmowej. Komenda usuwa log jedynie po stronie systemu DOMIQ.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.IDS.clearlog	1	Wykasowanie logu zdarzeń.



### 1. Przycisk do kasowania logu w aplikacji Remote

- W zakładce **Remote** dodaj element **Przycisk** i otwórz go w trybie edycji.
- **Etykieta:** *Kasowanie logu alarmu*
- Wybierz zakładkę **Krótko** -> **+Komenda**
- **Nazwa:** *C.IDS.clearlog*
- **Wartość:** *1*

## IDS.scan (tylko ETHM)

Komenda inicjująca pełny odczyt danych z centrali alarmowej. Odczytane zostaną następujące dane:

- wejścia
- wyjścia
- strefy
- ekspandery
- manipulatory
- użytkownicy
- pobranie logu zdarzeń

Wymuszenie ponownego skanowania jest przydatne po wprowadzeniu zmian w konfiguracji systemu alarmowego, np. dodanie nowych ekspanderów, użytkowników itd.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.IDS.scan	1	Uruchomienie skanowania.



### 1. Przycisk do uruchomienia skanowania w aplikacji Remote

- W zakładce **Remote** dodaj element **Przycisk** i otwórz go w trybie edycji.
- **Etykieta:** *Skan alarmu*
- Wybierz zakładkę **Krótko** -> **+Komenda**
- **Nazwa:** *C.IDS.scan*
- **Wartość:** *1*

## 12.4. DMX

Identyfikatory z grupy **DMX** są dedykowane do integracji systemu DOMIQ z urządzeniami wykorzystującymi protokół DMX512. Do integracji systemu DOMIQ ze sterownikami DMX512 wymagane jest użycie bramki sprzęgającej DOMIQ/Serial-4DX.

Więcej szczegółowych informacji nt. integracji urządzeń DOMIQ z DMX512 znajduje się w samouczku „**RGB LED przez DMX**” dostępnym do pobrania na naszej stronie internetowej [www.domiq.pl](http://www.domiq.pl), dział **Samouczki**.

### DMX.<kanał>

Identyfikator służący do zarządzania pojedynczym kanałem DMX-512. Nasz system umożliwia sterowanie maksymalnie 256 kanałami DMX, co przekłada się na możliwość sterowania 256 niezależnymi obwodami oświetlenia lub 85 obwodami oświetlenia RGB.

	Identyfikator	Wartość	Opis
<b>Stan</b>	DMX . <b>N</b> <b>N</b> - Numer kanału (0-255)	0-100	Aktualna wartość kanału DMX-512
<b>Zdarzenie</b>	E . DMX . <b>N</b>	0-100	Zmiana stanu kanału DMX-512
<b>Komenda</b>	C . DMX . <b>N</b>	0-100; ramp:xx xx - wartość rampy od 0 do 128. Parametr ramp jest opcjonalny.	Ustawienie wartości kanału DMX-512.
		stop	Stopuje ściemnianie/rozjaśnianie z rampą.
		on	Wysteruj jasność na wartość maksymalną.
		off	Wyłącz kanał (ustawia jasność 0%)
		toggle	Przełącza jasność pomiędzy 0 a 100%
		up	Skokowe zwiększenie jasności o 10%
		down	Skokowe zmniejszanie jasności o 10%
	dim	Komenda umożliwiająca ściemnianie/rozjaśnianie z użyciem przycisków ściennych i wirtualnych (w aplikacji Remote) w sposób znany z systemu LCN: krótkie naciśnięcie przycisku: włącz światło; kolejne krótkie naciśnięcie przycisku: wyłącz światło; długie naciśnięcie: rozjaśnianie; puszczenie przycisku: zatrzymuje zmianę jasności; kolejne długie naciśnięcie przycisku: ściemnianie; Patrz przykłady poniżej.	



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Sterowanie jasnością**

**Nazwa:** C . DMX . 1

**Wartość:** 100

Ustaw maksymalną jasność w kanale nr 1.

- **Nazwa:** C . DMX . 1

**Wartość:** 100 ; ramp : 5

Ustaw maksymalną jasność w kanale nr 1, użyj rampy z wartością 5.

- **Przełączenie stanu na przeciwny:**

**Nazwa:** C . DMX . 1

**Wartość:** toggle

Przełącz stan kanału nr 1 na przeciwny.

### 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie reagujące na wyłączenie oświetlenia.

**Kanał:** E . DMX . 1

**Dane:** 0

Kanał nr 1 został wyłączony.



### 3. Sterowanie z aplikacji DOMIQ/Remote

- **Kontrolka Światło RGBW**

Kontrolka **Światło RGBW** (zakładka **Remote**) to dedykowany element interfejsu służący do sterowania oświetleniem w przestrzeni barw RGB. Do realizacji takiego modelu sterowania są używane trzy kanały DMX. Natomiast jeżeli sterownik DMX wyposażony jest w osobny kanał dla światła białego, wówczas dostępny jest on jako czwarty kanał.

Na potrzeby przykładu założymy, że sterownik DMX dysponuje czterema kanałami DMX z numerami od 1 do 4, gdzie nr 1 to kolor czerwony, 2 - zielony, 3 - niebieski, a 4 - biały. Kanały te będą dostępne przy pomocy identyfikatorów od DMX . 1 do DMX . 4. Wówczas definicja kontrolki

**Światło RGBW** jest następująca:

**Etykieta:** Wpisz swój własny opis kontrolki.

**Czerwony:** DMX . 1

**Zielony:** DMX . 2

**Niebieski:** DMX . 3

**Biały:** DMX . 4

Uzupełnienie pola **Biały** jest opcjonalne i uzupełniamy je tylko wtedy, gdy sterownik DMX jest wyposażony w osobny kanał dla oświetlenia białego.

- **Kontrolka Przycisk**

Przy użyciu pojedynczej kontrolki typu **Przycisk** możliwe jest zrealizowane sterowanie oświetleniem z użyciem pojedynczego kanału DMX wg następującego schematu:

krótkie naciśnięcie przycisku -> włącz światło; kolejne krótkie naciśnięcie przycisku -> wyłącz światło; długie naciśnięcie -> rozjaśnianie; puszczenie przycisku -> zatrzymuje zmianę jasności; kolejne długie naciśnięcie przycisku -> ściemnianie; itd.

Na potrzeby przykładu przyjmijmy, że sterowany będzie kanał nr 1.

Definicja kontrolki **Przycisk:**

- Zakładka **Krótko** -> + **Komenda:**

**Nazwa:** C . DMX . 1

**Wartość:** toggle

Krótkie naciśnięcie realizuje funkcję włącz/wyłącz oświetlenie.

- Zakładka **Długo** -> + **Komenda:**

**Nazwa:** C . DMX . 1

**Wartość:** dim

Długie naciśnięcie realizuje zmianę jasności z rampą. Tak długo, jak przycisk jest przytrzymany, tak długo trwa zmiana jasności.

- Zakładka **Puść** -> + **Komenda:**

**Nazwa:** C . DMX . 1

**Wartość:** stop

Puszczenie przycisku zatrzymuje rampę.

- **Kontrolka Ściemniacz**

Kontrolka **Ściemniacz** daje możliwość sterowania wartością pojedynczego kanału DMX z użyciem suwaka. W polu **Kanał** należy wpisać identyfikator kanału DMX. Pozostałe pola kontrolki mogą pozostać z domyślnymi wartościami.

- **Kontrolka Przełącznik.**

Umożliwia sterowanie w trybie włącz/wyłącz. W polu **Kanał** należy wpisać identyfikator kanału DMX. Pola **Włącz** i **Wyłącz** mogą pozostać puste. Jeżeli chcesz, aby komenda była wykonywana z rampą, wówczas w polu **Parametr** wpisz: ramp : xx, gdzie xx to wartość rampy.

## DMX.scene

Identyfikator przeznaczony do sterowania scenami świetlnymi DMX. Sceny definiowane są w konfiguratorze w zakładce **DMX**.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.DMX.scene.S S – nazwa sceny przypisana w konfiguratorze (zakładka DMX).	load;ramp:xx Parametr ramp jest opcjonalny.	Ładuje wskazaną scenę. Domyślnie komenda używa rampy wskazanej przy konfiguracji sceny. Jeżeli jednak komenda zawiera parametr ramp, wówczas ma on pierwszeństwo. Pozwala to na jednorazowe wywołanie sceny z inną rampą, niż zadeklarowania w konfiguracji sceny.
		toggle;ramp:xx Parametr ramp jest opcjonalny.	Do wszystkich kanałów DMX przypisanych do danej sceny wysyła komendę toggle.
		stop	Stopuje ściemnianie/rozjaśnianie z rampą.
		off;ramp:xx Parametr ramp jest opcjonalny.	Wyłącza wszystkie kanały przypisane do danej sceny.
		dim	Działanie jak w przypadku pojedynczego kanału z tą różnicą, że sterowanie obejmuje wszystkie kanały przypisane do sceny.



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Wywołanie sceny**

**Nazwa:** C.DMX.scene.relax

**Wartość:** load

Załaduj scenę o nazwie *relax*.

- **Wyłączanie świateł przypisanych do sceny**

**Nazwa:** C.DMX.scene.relax

**Wartość:** off;ramp:7

Wyłącz wszystkie światła przypisane do sceny o nazwie *relax*, użyj ramp z wartością 7.

### 2. Sterowanie z aplikacji DOMIQ/Remote

W przypadku scen możliwe jest sterowanie z użyciem kontrolki **Przycisk** oraz **Przełącznik** w sposób opisany dla pojedynczych kanałów DMX. Należy jedynie pamiętać, aby w edycji kontrolki odwoływać się do identyfikatora sceny.

Ponadto przy użyciu kontrolki **Przycisk** można wywoływać wybrane sceny w następujący sposób:

- Zakładka **Krótko** -> **+ Komenda**

**Nazwa:** C.DMX.scene.relax

**Wartość:** load;ramp:5

- Analogicznie można utworzyć więcej przycisków i zgrupować je w jeden element interfejsu z użyciem kontrolki **Grupa Przycisków**.

## 12.5. MODBUS

Identyfikatory z grupy **MODBUS** są dedykowane do integracji systemu DOMIQ z urządzeniami wykorzystującymi protokół MODBUS.

Nazwy identyfikatorów z grupy **MODBUS** są tworzone na podstawie nazw nadanych podczas konfigurowania parametrów integracji w zakładce **MODBUS**.

	Identyfikator	Wartość	Opis
<b>Stan</b>	MODBUS.I.U.R I - Nazwa interfejsu U - Nazwa urządzenia R - Nazwa rejestru	Liczba	Odczyt aktualnej zawartości rejestru urządzenia MODBUS
<b>Zdarzenie</b>	E.MODBUS.I.U.R	Liczba	Informacja o zmianie zawartości rejestru.
<b>Komenda</b>	C.MODBUS.I.U.R	Liczba	Zapis wartości do rejestru MODBUS.



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Ustawienie wartości rejestru**

**Nazwa:** C.MODBUS.int.piec.tempzadana

**Wartość:** 23

Przypisz wartość 23 do rejestru: MODBUS.int.piec.tempzadana.

### 2. Zdarzenia na zmianę wartości rejestru

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

**Kanał:** E.MODBUS.tcp.licznik.energia

**Dane:** 125.7

Rejestr MODBUS o nazwie *energia*, w urządzeniu *licznik*, przypisanym do interfejsu *tcp* osiągnął wartość 125.7.

Praktyczne przykłady zastosowania identyfikatorów MODBUS znajdziesz w samouczkach „**Licznik energii i MODBUS**”, „**Stacja pogodowa MODBUS**” oraz „**MODBUS/TCP i sterowniki Beckhoff**” gdzie kolejno przedstawiliśmy proces integracji urządzeń DOMIQ z cyfrowym licznikiem energii elektrycznej, stacją pogodową oraz modułarnym sterownikiem PLC marki Beckhoff. Samouczki są do pobrania z naszej strony internetowej [www.domiq.pl](http://www.domiq.pl), dział **Samouczki**.

## 12.6. EXP

Identyfikatory z grupy **EXP** przeznaczone są do sterowania i integracji ekspanderów Satel podłączonych do modułu DOMIQ/Expander. Poniższa tabela zawiera przegląd wszystkich identyfikatorów **EXP** i ich główne cechy. Szczegółowy opis poszczególnych identyfikatorów wraz z przykładami zastosowania znajduje się w dalszej części podrozdziału.

Grupa	Skrócony opis	Stan	Zdarzenia	Komenda
EXP.output	Sterowanie i stan pojedynczego przekaźnika w ekspanderze.	+	+	+
EXP.outputs	Sterowanie kilkoma przekaźnikami w obrębie jednego ekspandera.	-	-	+
EXP.group	Sterowanie wszystkimi przekaźnikami przypisanymi do danej grupy.	-	-	+
EXP.all	Sterowanie jednocześnie wszystkimi przekaźnikami w instalacji.	-	-	+
EXP.input	Stan pojedynczego wejścia ekspandera oraz zdarzenia informujące o zmianie stanu.	+	+	-
EXP.shutter	Sterowanie roletami podłączonymi do modułu INT-IORS	-	-	+
EXP.button	Zdarzenie emitowane przy naciśnięciu przycisków podłączonych do wejść modułu INT-IORS. Rozróżniane jest długie i krótkie naciśnięcie przycisku oraz jego puszczenie.	-	+	-
EXP.imp	Umożliwia załączanie przekaźnika na określony odcinek czasu.	-	-	+
EXP.scan	Komenda inicjująca skanowanie magistrali w poszukiwaniu przyłączonych modułów INT-IORS	-	-	+

## 12. Identyfikatory

### EXP.output

Identyfikator EXP.output przeznaczony jest do zarządzania pojedynczym przekaźnikiem.

	Identyfikator	Wartość	Opis
Stan	EXP.output.E.O E- Numer ekspandera o - Numer wyjścia	0	Przełącznik wyłączony
		1	Przełącznik włączony
Zdarzenie	E.EXP.output.E.O	0	Przełącznik został wyłączony
		1	Przełącznik został włączony
Komenda	C.EXP.output.E.O	1/on	Włącz
		0/off	Wyłącz
		toggle	Przełącz na stan przeciwny



#### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Włączenie przekaźnika**

**Nazwa:** C.EXP.output.1.1

**Wartość:** 1

Włącz pierwszy przekaźnik w ekspanderze z adresem 1.

- **Przełączenie przekaźnika**

**Nazwa:** C.EXP.output.1.1

**Wartość:** toggle

Przełącz pierwszy przekaźnik w ekspanderze z adresem 1.

#### 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie reagujące na załączenie przekaźnika.

**Kanał:** E.EXP.output.5.2

**Dane:** 1

Przełącznik nr 2 w ekspanderze o adresie 5 został włączony.

#### 3. Sterowanie z aplikacji DOMIQ/Remote

- **Kontrolka Przełącznik**

W polu **Kanał** wpisz identyfikator wyjścia np. EXP.output.3.1.

#### 4. Sterowanie z wizualizacji

- **Kontrolka Światło**

W polu **Kanał** wpisz: EXP.output.3.1

- Analogicznie można również sterować z użyciem kontrolki **Przełącznik**.



## EXP.outputs

Identyfikator `EXP.outputs` służy do sterowania jednocześnie kilkoma przekaźnikami w obrębie jednego ekspandera. Składnia polecenia przypomina tą znaną z identyfikatora `LCN.relays`.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<code>C.EXP.outputs.E</code> E - Numer ekspandera	0/1/T/-	Komenda składająca się z 8 znaków, gdzie poszczególne znaki odpowiadają kolejnym przekaźnikom. 0 - wyłącz przekaźnik 1 - włącz przekaźnik T - zmień stan przekaźnika na przeciwny - - nie zmieniaj stanu



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- Nazwa:** `C.EXP.outputs.1`  
**Wartość:** `--11----`  
 Włącz przekaźniki nr 3 i 4 w ekspanderze o adresie 1. Pozostałe przekaźniki pozostaw bez zmian.
- Nazwa:** `C.EXP.outputs.1`  
**Wartość:** `TT1100--`  
 W ekspanderze o adresie 1: przełącz dwa pierwsze przekaźniki, kolejne dwa włącz, przekaźniki nr 5 i 6 wyłącz, ostatnie dwa pozostaw bez zmian.

### 2. Sterowanie z wizualizacji

- Kontrolka Przycisk**  
 W polu **Komenda** wpisz: `EXP.outputs.1=TT1100--`  
 Każdorazowe naciśnięcie przycisku wykona nast. akcję: w ekspanderze o adresie 1: przełącz dwa pierwsze przekaźniki, kolejne dwa włącz, przekaźniki nr 5 i 6 wyłącz, ostatnie dwa pozostaw bez zmian.

## 12. Identyfikatory

### EXP.group

Identyfikator `EXP.group` przeznaczony jest do sterowania wszystkim przekaźnikami przynależnymi do danej grupy. Grupy można zdefiniować w zakładce **Ekspandery** -> **Grupy**.

Sterowanie grupami jest bardzo przydatne do tworzenia globalnych reguł sterowania, np. wyłączenie oświetlenia w momencie uzbrojenia alarmu. Wówczas możliwym staje się sterowanie wieloma obwodami przy użyciu pojedynczej komendy!

Przykłady zastosowania:

- grupa dla oświetlenia w określonych pomieszczeniach;
- grupa dla oświetlenia zewnętrznego;
- grupa dla gniazdek elektrycznych;
- grupa dla zaworów np. wody/gazu/ogrzewania;

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<code>C.EXP.group.G</code> G - numer grupy	toggle	Przełącz na przeciwny
		on/1	Włącz
		off/0	Wyłącz



#### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Nazwa:** `C.EXP.group.1`  
**Wartość:** 1  
Włącz przekaźniki przypisane do grupy nr 1.
- **Nazwa:** `C.EXP.group.1`  
**Wartość:** toggle  
Przełącz wszystkie przekaźniki przynależne do grupy nr 1.

## EXP.all

Identyfikator `EXP.all` przeznaczony jest do jednoczesnego sterowania wszystkim przekaźnikami w instalacji (we wszystkich podłączonych modułach INT-IORS).

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<code>C.EXP.all</code>	<code>toggle</code>	Przełącz na przeciwny
		<code>on/1</code>	Włącz
		<code>off/0</code>	Wyłącz



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- Nazwa:** `C.EXP.all`  
**Wartość:** `on`  
 Włącz wszystkie przekaźniki w instalacji
- Nazwa:** `C.EXP.all`  
**Wartość:** `0`  
 Wyłącz wszystkie przekaźniki w instalacji.

## EXP.input

Identyfikator `EXP.input` odwołuje się do stanu pojedynczego wejścia ekspandera oraz emitowane jest zdarzenie informujące o zmianie stanu wejścia.

	Identyfikator	Wartość	Opis
<b>Stan</b>	<code>EXP.input.E.O</code> E- Numer ekspandera o - Numer wejścia	0	Wejście rozwarte
		1	Wejście zwarte
<b>Zdarzenie</b>	<code>E.EXP.input.E.O</code>	0	Wejście zostało rozwarte
		1	Wejście zostało zwarte



### 1. Zdarzenia informujące o zmianie stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Kanał:** `E.EXP.input.1`  
**Dane:** 1  
 Wejście nr 1 zmieniło wartość na 1. Wejściem alarmowym może być np. czujka PIR, kontaktron itd. Jeżeli założymy, że naszym wejściem jest kontaktron zamontowany na ramie okiennej, wówczas powyższy przykład należy rozumieć jako: Otwarto okno nr ... .

## 12. Identyfikatory

### EXP.shutter

Identyfikator służący do sterowania roletami podłączonymi do modułów INT-IORS. Każda roleta sterowana jest przez parę przekaźników: pierwszy przekaźnik w parze odpowiada za zasilanie silnika, drugi steruje kierunkiem. Stan poszczególnych rolet dostępny jest jako stan przekaźników obsługujących daną roletę: oba przekaźniki włączone oznaczają ruch rolety w dół, natomiast gdy włączony jest tylko pierwszy przekaźnik, wówczas roleta porusza się do góry.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.EXP.shutter.E.R E - numer ekspandera R - numer rolety Dopuszczalne wartości: 1-4 lub all (sterowanie wszystkim czterema roletami jednocześnie)	up/1	Ruch rolety w górę.
		down/3	Ruch rolety w dół.
		stop/0	Zatrzymanie rolety.
		toggleup/4	Każdorazowe wywołanie komendy powoduje zmianę pomiędzy dwoma stanami: ruch w górę lub zatrzymanie silnika. Komenda jest szczególnie przydatna do sterowania roletami z przycisków ściennych, gdy do dyspozycji mamy tylko 2 klawisze. Ta komenda jest obsługiwana wyłącznie w trybie pojedynczej rolety.
		toggledown/5	Każdorazowe wywołanie komendy powoduje zmianę pomiędzy dwoma stanami: ruch w dół lub zatrzymanie silnika. Komenda jest szczególnie przydatna do sterowania roletami z przycisków ściennych, gdy do dyspozycji mamy tylko 2 klawisze. Ta komenda jest obsługiwana wyłącznie w trybie pojedynczej rolety.
toggle	Atrybut <code>time</code> nie jest obsługiwany!	Komenda umożliwiająca jednoprzyciskowe sterowanie roletą. Każde kolejne wywołanie komendy zmienia stan rolety wg następującego schematu: w górę - stop - w dół - stop - w górę itd. Jest to szczególnie przydatne wszędzie tam, gdzie do dyspozycji jest tylko 1 klawisz sterujący roletą.	

**WAŻNE:** Każda komenda sterująca ruchem rolety (poza `toggle`) akceptuje dodatkowy atrybut, który umożliwia wysterowanie rolety na określony odcinek czasu. Wartość może być również liczbą ułamkową. Ogólna składnia polecenia: `C.EXP.shutter.<moduł>.<roleta>=<komenda>;<czas>`. Więcej szczegółów w przykładach poniżej.

Każda komenda oprócz `toggle` posiada swój numeryczny odpowiednik. Jest to przydatne w tych przypadkach, gdy nie ma możliwości tekstowego wywołania danej komendy, np. integracja z zewnętrznym systemem daje możliwość wysyłania jedynie wartości numerycznych.



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Nazwa:** C.EXP.shutter.3.1  
**Wartość:** up  
Rozpocznij ruch w górę rolety nr 1 w module nr 3.
- **Ruch rolety z interwałem czasowym**  
**Nazwa:** C.EXP.shutter.3.1  
**Wartość:** up;3  
Wykonaj ruch w górę przez 3 sekundy dla rolety nr 1 w module nr 3.
- **Nazwa:** C.EXP.shutter.3.4  
**Wartość:** toggledown  
Przełącz stan na przeciwny dla rolety nr 4 w module nr 3. Stan zostanie przełączony między ruchem w dół lub zatrzymaniem rolety.
- Sterowanie wszystkimi roletami w obrębie modułu INT-IORS  
**Nazwa:** C.EXP.shutter.3.all  
**Wartość:** down  
Opuść wszystkie rolety w module nr 3.

### 2. Sterowanie z aplikacji DOMIQ/Remote

- **Z użyciem kontrolki Roleta**  
Dodaj kontrolkę **Roleta**, a następnie w polu **Kanał** wpisz identyfikator rolety, np. EXP.shutter.3.1
- **Sterowanie jedнопrzyciskowe z użyciem kontrolki Przycisk.**  
Na potrzeby przykładu przyjęliśmy roletę nr 1 w module nr 3.
  - Dodaj **Przycisk**.
  - W zakładce **Krótko** -> **+ Komenda**  
**Nazwa:** C.EXP.shutter.3.1  
**Wartość:** toggle

## 12. Identyfikatory

### EXP.button

Przy użyciu tego identyfikatora rozgłaszane są zdarzenia dotyczące naciskania przycisków podłączonych do wejść modułów INT-IORS.

	Identyfikator	Wartość	Opis
Zdarzenie	E.EXP.button.E.P E - numer ekspandera P - numer wejścia, do którego podłączony jest przycisk.	short	Zdarzenie informujące o krótkim (poniżej sekundy) naciśnięciu przycisku.
		long	Zdarzenie informujące o długim (powyżej sekundy) naciśnięciu przycisku.
		release	Zdarzenie informujące o puszczeniu przycisku.



#### 1. Zdarzenia informujące o naciśnięciu przycisku

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- **Kanał:** E.EXP.button.1.1  
**Dane:** long  
Długie naciśnięcie przycisku podłączonego do wejścia nr 1 w module nr 1.
- **Kanał:** E.EXP.button.1.1  
**Dane:** short  
Krótkie naciśnięcie przycisku podłączonego do wejścia nr 1 w module nr 1.
- **Kanał:** E.EXP.button.1.1  
**Dane:** release  
Zwolnienie przycisku podłączonego do wejścia nr 1 w module nr 1.

### EXP.imp

Komenda umożliwiająca wysterowanie przekaźnika w module INT-IORS na określony odcinek czasu. Dzięki temu z łatwością można zrealizować sterowanie bramami/furtkami i innymi obwodami, gdzie potrzeba wysterować przekaźnik na określony odcinek czasu.

	Identyfikator	Wartość	Opis
Komenda	C.EXP.imp.E.O E - numer ekspandera O - numer przekaźnika	czas wyrażony w milisekundach	Włączenie przekaźnika na wskazany odcinek czasu.



#### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Nazwa:** C.EXP.imp.3.1  
**Wartość:** 500  
Włącz przekaźnik nr 1 w module nr 3 na pół sekundy (500ms).
- **Nazwa:** C.EXP.imp.3.1  
**Wartość:** 10000  
Włącz przekaźnik nr 1 w module nr 3 na 10 sekund.

## EXP.scan

Komenda inicjująca skanowanie magistrali w poszukiwaniu podłączonych ekspanderów. Skanowanie magistrali jest uruchamiane automatycznie przy starcie modułu Base, jeśli jest podłączony moduł DOMIQ/Expander. Skanowanie ręczne należy wykonać w dwóch przypadkach:

- podłączenie nowych ekspanderów do magistrali;
- zmiana adresu jednego lub wielu ekspanderów;

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.EXP.scan	1	Uruchomienie skanowania magistrali ekspanderów.

# 12.7. DALI

Identyfikatory z grupy **DALI** służą do sterowania oświetleniem DALI za pomocą modułu DOMIQ/Light. Każdy identyfikator realizuje ściśle określoną funkcjonalność powiązaną z interfejsem DALI.

Identyfikatory z grupy **DALI** można zasadniczo podzielić ze względu na obszar zastosowania. Pierwszą podgrupą są identyfikatory typu `DALI.1.all`, które są przeznaczone do sterowania jednocześnie całą instalacją DALI.

Drugą podgrupę stanowią identyfikatory typu `DALI.1.evlg`, które są dedykowane do sterowania pojedynczym balastem DALI oraz informowania o stanie balastu i jego zmianach.

Trzecią grupą są identyfikatory `DALI.1.group`, które są wykorzystywane do sterowania grupami oświetlenia DALI.

Poniższa tabela zawiera przegląd typów identyfikatorów DALI i ich główne cechy. Szczegółowy opis poszczególnych identyfikatorów wraz z przykładami zastosowania znajduje się w dalszej części podrozdziału.

Grupa	Skrócony opis	Stan	Zdarzenia	Komenda
<code>DALI.1.evlg</code>	Grupa identyfikatorów przeznaczona do sterowania pojedynczym balastem DALI.	+	+	+
<code>DALI.1.group</code>	Grupa identyfikatorów przeznaczona do sterowania grupami oświetlenia.	-	-	+
<code>DALI.1.all</code>	Grupa identyfikatorów przeznaczona do sterowania jednocześnie całą instalacją DALI.	-	-	+



## DALI.1.evg

Identyfikator DALI.1.evg wykorzystywany jest do zarządzania pojedynczym balastem DALI.

	Identyfikator	Wartość	Opis
<b>Stan</b>	DALI.1.evg. <b>A</b> <b>A</b> - Adres balastu DALI.	0-100	Bieżąca jasność.
	DALI.1.evg. <b>A</b> .on	0/1	Informacja o tym, czy balast jest włączony.
	DALI.1.evg. <b>A</b> .powerfail	0/1	Informacja o wystąpieniu awarii zasilania.
	DALI.1.evg. <b>A</b> .lampfail	0/1	Informacja o wystąpieniu awarii balastu.
	DALI.1.evg. <b>A</b> .scene	0-15	Informacja o aktualnie załadowanej scenie.
<b>Zdarzenie</b>	<b>E</b> .DALI.1.evg. <b>A</b>	0-100	Zdarzenie informujące o zmianie bieżącej jasności.
	<b>E</b> .DALI.1.evg. <b>A</b> .on	0-1	Zdarzenie informujące o załączeniu lampy.
	<b>E</b> .DALI.1.evg. <b>A</b> .lampfail	0-1	Zdarzenie informujące o wystąpieniu awarii lampy.
	<b>E</b> .DALI.1.evg. <b>A</b> .powerfail	0-1	Zdarzenie informujące o wystąpieniu awarii zasilania.
<b>Komenda</b>	<b>C</b> .DALI.1.evg. <b>A</b>	0-100	Ustaw jasność w %.
		on	Załączenie wybranej lampy. Ustawiona zostanie jasność wg parametru <b>Maksimum</b> .
		off	Wyłączenie wybranej lampy.
		max	Ustaw jasność na maksimum.
		min	Ustaw jasność na minimum.
		up	Skokowe rozjaśnianie. Skok uzależniony jest od parametru <b>Skok</b> .
		down	Skokowe ściemnianie. Skok uzależniony jest od parametru <b>Skok</b> .
		stepup	Skokowe rozjaśnianie o 1. Komenda nie powoduje załączenia oświetlenia.
		stepdown	Skokowe ściemnianie o 1. Komenda nie powoduje wyłączenia oświetlenia.
		onandstepup	Skokowe rozjaśnianie o 1. Wywołanie tej komendy, gdy oświetlenie jest wyłączone spowoduje jego załączenie i ustawienie jasności na minimum.
		stepdownandoff	Skokowe ściemnianie o 1. Gdy jasność osiągnie minimum, komenda powinna wyłączyć oświetlenie.
		load: <b>xx</b>	Załadowanie wybranej sceny, gdzie <b>xx</b> oznacza numer sceny.
		save: <b>xx</b>	Zapisanie bieżącej jasności do sceny, gdzie <b>xx</b> oznacza numer sceny.
		prev	Załadowanie sceny poprzedniej względem ostatnio załadowanej. Jeżeli wartość dojdzie do 0, wówczas załadowana zostanie scena nr 15.
next next; limit: <b>xx</b>	Załadowanie sceny następnej względem ostatnio załadowanej. Jeżeli wartość dojdzie do 15 lub wartości określonej przez parametr limit, wówczas załadowana zostanie scena nr 0.		

## 12. Identyfikatory

Komenda	C.DALI.1.evlg.A	fadetime:xx	Komenda ustawiająca rampę w danym balaście, gdzie <b>xx</b> oznacza wartość rampy.
		0-100;fadetime:xx	Domyślnie rampa jest zapisana w pamięci balastu i wówczas każda komenda zmieniająca jasność wykonywana jest z daną rampą. W przypadku tej komendy, jasność zmieniana jest z rampą jednorazowo. Po każdej komendzie rampa jest zerowana. <b>xx</b> oznacza wartość rampy.
		faderate:xx	Komenda ustawiająca skok, o który zostanie zmieniona jasność przy wywołaniu komend <code>up</code> lub <code>down</code> . <b>xx</b> oznacza wartość skoku. Identyfikatorowy odpowiednik komendy do ustawiania <b>Skoku</b> dostępnego w oknie <b>Parametry</b> .
		dtr:xx	Zapisanie wartości w rejestrze dtr. „ <b>xx</b> ” oznacza zapisywaną wartość.
		toggle toggle:xx toggle:last toggle;load:xx	<ul style="list-style-type: none"> <li>• <code>toggle</code> - przełącza jasność między 0 a 100%.</li> <li>• <code>toggle:xx</code>, gdzie <code>xx</code> to zadana jasność. Komenda przełącza jasność pomiędzy 0 a wartością zadaną.</li> <li>• <code>toggle:last</code> - przełącza jasność między 0% a ostatnią wartością niezerową.</li> <li>• <code>toggle;load:xx</code> - gdzie <code>xx</code> to numer sceny świetlnej przypisanej do balastu. Przełącza wartość pomiędzy 0%, a jasnością zapisaną we wskazanej scenie.</li> </ul> <p><b>Wszystkie wersje komendy <code>toggle</code> uwzględniają rampę wprowadzoną we właściwościach balastu</b></p>
		tap	Przełącza stan na przeciwny, nie uwzględnia rampy - komenda załącza/wyłącza balast natychmiast.
		dim	Komenda umożliwiająca ściemnianie/rozjaśnianie z użyciem przycisków ściennych i wirtualnych (w aplikacji Remote) w sposób znany z systemu LCN: krótkie naciśnięcie przycisku: włącz światło; kolejne krótkie naciśnięcie przycisku: wyłącz światło; długie naciśnięcie: rozjaśnianie; puszczenie przycisku: zatrzymuje zmianę jasności; kolejne długie naciśnięcie przycisku: ściemnianie;
		reset	Wyzerowanie ustawień zmiennych zapisanych w pamięci balastu DALI.



## 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Włączenie balastu**

**Nazwa:** C.DALI.1.evg.5

**Wartość:** on

Włącz balast o adresie 5. Jasność zostanie ustawiona zgodnie z parametrem **Maksimum** danego balastu.

- **Załadowanie sceny świetlnej**

**Nazwa:** C.DALI.1.evg.10

**Wartość:** load:2

Załaduj scenę nr 2 w balaście o adresie 10.

- **Zapisanie sceny świetlnej**

**Nazwa:** C.DALI.1.evg.10

**Wartość:** save:2

Zapisz bieżącą jasność jako scenę nr 2 w balaście o adresie 10.

- **Ustawienie rampy**

**Nazwa:** C.DALI.1.evg.10

**Wartość:** fadetime:10

Ustaw parametr **Rampa** w lampie o adresie 10 na wartość 10.

- **Ustawienie jasności oświetlenia z wybraną rampą**

**Nazwa:** C.DALI.1.evg.1

**Wartość:** 100;fadetime:5

Wysteruj balast o adresie 1 na wartość maksymalną z rampą 5.

- **Przełączanie stanu na przeciwny**

**Nazwa:** C.DALI.1.evg.1

**Wartość:** toggle:70

Komenda realizująca przełączanie jasności między wartością 0 a 70%.

## 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Zdarzenie reagujące na zmianę jasności.

**Kanał:** E.DALI.1.evg.0

**Dane:** 50

Balast o adresie 0 został wysterowany na 50%.



### 3. Sterowanie z aplikacji DOMIQ/Remote

- **Kontrolka Ściemniacz**  
Umożliwia sterowanie jasnością balastu przy pomocy suwaka. Ten rodzaj sterowania wymaga, aby parametr **Rampa** we właściwościach balastu miał wartość 0.  
W polu **Kanał** wpisz identyfikator balastu np. DALI . 1 . evg . 10
- **Kontrolka Przycisk**  
Przy użyciu pojedynczej kontrolki typu **Przycisk** możliwe jest zrealizowane sterowanie balastem DALI wg następującego schematu:  
krótkie naciśnięcie przycisku -> włącz światło; kolejne krótkie naciśnięcie przycisku -> wyłącz światło; długie naciśnięcie -> rozjaśnianie; puszczenie przycisku -> zatrzymuje zmianę jasności; kolejne długie naciśnięcie przycisku -> ściemnianie; itd.  
Na potrzeby przykładu przyjmijmy, że sterowany będzie balast nr 10.  
Definicja kontrolki **Przycisk**:
  - Zakładka **Krótko** -> **+ Komenda**:  
**Nazwa**: C . DALI . 1 . evg . 10  
**Wartość**: tap  
Krótkie naciśnięcie realizuje funkcję włącz/wyłącz oświetlenie.
  - Zakładka **Długo** -> **+ Komenda**:  
**Nazwa**: C . DALI . 1 . evg . 10  
**Wartość**: dim  
Długie naciśnięcie realizuje zmianę jasności z rampą. Tak długo, jak przycisk jest przytrzymany, tak długo trwa zmiana jasności.
  - Zakładka **Puść** -> **+ Komenda**:  
**Nazwa**: C . DALI . 1 . evg . 10  
**Wartość**: stop  
Puszczenie przycisku zatrzymuje rampę.
- **Kontrolka Przełącznik**.  
Umożliwia sterowanie w trybie włącz/wyłącz. W polu **Kanał** należy wpisać identyfikator balastu DALI. Pola **Włącz** i **Wyłącz** mogą pozostać puste. Jeżeli chcesz, aby komenda była wykonywana z rampą, wówczas w polu **Parametr** wpisz: ramp : xx, gdzie xx to wartość rampy.

## DALI.1.group

Identyfikatory z grupy `DALI.1.group` służą do sterowania grupami oświetlenia DALI. Zestaw dostępnych komend jest identyczny, jak w przypadku grupy `DALI.1.evq` z wyłączeniem komend: `dim`, `toggle` (wszystkie warianty), `tap`, `prev` i `next`.

Komendy są rozsyłane jednocześnie do wszystkich balastów przynależnych do danej grupy. Identyfikatory z tej grupy nie generują zdarzeń oraz nie mają stanu. Jednakże wysłanie komendy do grupy powoduje wygenerowanie odpowiednich zdarzeń w obrębie balastów przynależnych do tej grupy oraz aktualizację ich stanu wg odebranej komendy.

## DALI.1.all

Identyfikatory z tej grupy dedykowane są do sterowania całą instalacją DALI jednocześnie. Komendy wysyłane przy pomocy tych identyfikatorów wykonywane są przez wszystkie balasty dostępne w instalacji, niezależnie czy dany balast posiada przypisany adres czy też nie.

Zestaw dostępnych komend jest identyczny, jak w przypadku grupy `DALI.1.evq` z wyłączeniem komend: `dim`, `toggle` (wszystkie warianty), `tap`, `prev` i `next`.

Komendy są rozsyłane jednocześnie do wszystkich balastów dostępnych w instalacji. Zatem wysłanie np. komendy `C.DALI.1.all=on` spowoduje włączenie wszystkich balastów w instalacji itd. Ta grupa identyfikatorów również nie generuje zdarzeń oraz nie posiada własnego stanu. Jednakże wysłanie komendy do wszystkich balastów spowoduje wygenerowanie odpowiednich zdarzeń w obrębie poszczególnych lamp i zaktualizowanie stanu wg otrzymanej komendy. Przykładowo, jeżeli wysłana zostanie komenda `C.DALI.all=on`, wówczas każdy balast osobno wygeneruje zdarzenie informujące o jego załączeniu oraz zaktualizuje stan jasności.

# 12.8. SHELLY

Identyfikatory SHELLY służą do zarządzania urządzeniami Shelly. Wszystkie zaprezentowane komendy oraz identyfikatory do odczytu stanu są uogólnioną formą zapisu zawierającą fragmenty, które należy zastąpić danymi dotyczącymi konkretnego modułu Shelly. Są one ujęte w nawiasy ostre -<dowolna\_nazwa>.

**Przykład:** SHELLY.<nazwa>.relay.<0-1>=<komenda>.

Numeracja przekaźników w Shelly zaczyna się od 0. Zatem gdy moduł posiada tylko jeden przekaźnik, wówczas jego indeks to 0.

Analogicznie przedstawia się numeracja wejść binarnych. Wejścia binarne umożliwiają podłączenie tradycyjnych łączników oświetleniowych (również dzwonekowych), a także innych urządzeń, które są w stanie zwierać styki wejścia.

W module Base zaimplementowaliśmy obsługę następujących urządzeń Shelly

### Generacja 1:

1. 1/1PM/1L
2. 2/2.5
3. Plug/PlugS
4. Dimmer 1/2
5. RGBW2 (tylko tryb White)
6. Uni
7. TRV
8. Vintage
9. H&T
10. Smoke
11. Flood
12. Door/Window 1/2
13. Motion
14. Sense
15. Button1
16. I3
17. Gas
18. EM
19. 3EM

### Generacja 2:

1. Plus 1/1 PM
2. Plus 2 PM
3. Plus I4
4. Pro 1/1 PM
5. Pro 2/2 PM
6. Pro 4 PM

## Generacja 1

### 1/1PM/1L

Moduł wykonawczy wyposażony w 1 przekaźnik oraz 1 wejście przyciskowe. Wersja PM wyposażona jest w pomiar mocy i raportowanie zużycia energii.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.relay.0	0/off	Wyłącz.
		1/on	Włącz.
		toggle	Przełącz.
<b>Stan</b>	SHELLY.<nazwa>.relay.0	0-1	Aktualny stan przekaźnika.
	SHELLY.<nazwa>.input.0	0-1	Aktualny stan wejścia binarnego.
	SHELLY.<nazwa>.longpush.0	0-1	Informacja o długim naciśnięciu przycisku.
	SHELLY.<nazwa>.relay.0.power (tylko wersja PM)	dowolna liczba	Bieżący pobór mocy.
	SHELLY.<nazwa>.relay.0.energy (tylko wersja PM)	dowolna liczba	Zużycie energii [W/m].
	SHELLY.<nazwa>.temperature (tylko wersja PM)	dowolna liczba	Wewnętrzna temperatura urządzenia.
	SHELLY.<nazwa>.overtemperature (tylko wersja PM)	0-1	Informacja o przegrzeniu urządzenia (wtedy wartość 1).
	SHELLY.<nazwa>.temperature_status (tylko wersja PM)	normal/ high/ veryhigh	Status temperatury urządzenia.
	SHELLY.<nazwa>.overpower_value (tylko wersja PM)	dowolna liczba	Jeżeli jest przeciążenie przekaźnika, ten parametr zawiera informację o ile watów przekroczone limit.
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .
	E.SHELLY.<nazwa>.input_event.0	S/L	Informacja o zwarcie wejścia binarnego: S - krótkie L - długie <u>Tylko, gdy wejście w trybie Momentary</u>

## 12. Identyfikatory

### 2.5

Moduł wyposażony w dwa przekaźniki z pomiarem mocy i dwa wejścia binarne. Może pracować w trybie roletowym lub jako dwa niezależne przekaźniki.

#### Tryb roletowy

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.roller.0	up	W górę
		down	W dół.
		stop	Stop.
		rc	Kalibracja rolety.
		0-100	Ustawienie rolety na określoną pozycję. 100 - pełne otwarcie 0 - pełne zamknięcie
<b>Stan</b>	SHELLY.<nazwa>.roller.0	up/down/ stop	Aktualny stan rolety.
	SHELLY.<nazwa>.roller.0.pos	0-100	Aktualna pozycja rolety. Opcja dostępna tylko po kalibracji. Wartość -1 oznacza błędną pozycję (brak kalibracji).
	SHELLY.<nazwa>.input.<0-1>	0-1	Informacja o stanie wejścia binarnego.
	SHELLY.<nazwa>.roller.0.power	dowolna liczba	Bieżący pobór mocy.
	SHELLY.<nazwa>.roller.0.energy	dowolna liczba	Zużycie energii [W/m].
	SHELLY.<nazwa>.temperature	dowolna liczba	Wewnętrzna temperatura urządzenia.
	SHELLY.<nazwa>.overtemperature	0-1	Informacja o przegrzeniu urządzenia (wtedy wartość 1).
	SHELLY.<nazwa>.temperature_status	normal/ high/ veryhigh	Status temperatury urządzenia.
	SHELLY.<nazwa>.voltage	dowolna liczba	Bieżące napięcie modułu.
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .
	E.SHELLY.<nazwa>.input_event.0	S/L	Informacja o zwarcie wejścia binarnego: S - krótkie L - długie <u>Tylko, gdy wejście w trybie Momentary</u>



## Tryb przekaźnikowy

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.relay.<0-1>	0/off	Wyłącz.
		1/on	Włącz.
		toggle	Przełącz.
<b>Stan</b>	SHELLY.<nazwa>.relay.<0-1>	0-1	Stan przekaźnika.
	SHELLY.<nazwa>.input.<0-1>	0-1	Stan wejścia binarnego.
	SHELLY.<nazwa>.longpush.<0-1>	0-1	Informacja o długim naciśnięciu przycisku.
	SHELLY.<nazwa>.relay.<0-1>.power	dowolna liczba	Bieżący pobór mocy.
	SHELLY.<nazwa>.relay.<0-1>.energy	dowolna liczba	Zużycie energii [W/m].
	SHELLY.<nazwa>.relay.<0-1>.overpower_value	dowolna liczba	Jeżeli jest przeciążenie przekaźnika, ten parametr zawiera informację o ile watów przekroczono limit.
	SHELLY.<nazwa>.temperature	dowolna liczba	Wewnętrzna temperatura urządzenia.
	SHELLY.<nazwa>.overtemperature	0-1	Informacja o przegrzeniu urządzenia (wtedy wartość 1).
	SHELLY.<nazwa>.temperature_status	normal/ high/ veryhigh	Status temperatury urządzenia.
	SHELLY.<nazwa>.voltage	dowolna liczba	Bieżące napięcie sieciowe modułu.
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .
	E.SHELLY.<nazwa>.input_event.<0-1>	S/L	Informacja o zwarciu wejścia binarnego: S - krótkie L - długie <u>Tylko, gdy wejście w trybie Momentary</u>

## 12. Identyfikatory

### Plug/PlugS

Sterowane gniazdko z pomiarem mocy.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.relay.0	0/off	Wyłącz.
		1/on	Włącz.
		toggle	Przełącz.
<b>Stan</b>	SHELLY.<nazwa>.relay.0	0-1	Aktualny stan przekaźnika.
	SHELLY.<nazwa>.relay.0.power	dowolna liczba	Bieżący pobór mocy.
	SHELLY.<nazwa>.relay.0.energy	dowolna liczba	Zużycie energii [W/m].
	SHELLY.<nazwa>.temperature	dowolna liczba	Wewnętrzna temperatura urządzenia.
	SHELLY.<nazwa>.overtemperature	0-1	Informacja o przegrzeniu urządzenia (wtedy wartość 1).
	SHELLY.<nazwa>.overpower_value	dowolna liczba	Jeżeli jest przeciążenie przekaźnika, ten parametr zawiera informację o ile watów przekroczono limit.
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .

## Dimmer 1/2

Moduł do sterowania oświetleniem z opcją ściemniania. Moduł wyposażony jest także w wejście binarne.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.light.0	0-100	Ustawienie określonej jasności.
		on	Włącz.
		off	Wyłącz.
		toggle	Przełącz.
		<komenda>; ramp:<rampa>	Wszystkie powyższe komendy z opcją użycia rampy. Rampa określa czas przejścia ze stanu obecnego do zadanego. Maksymalna wartość rampy 5 [s]
<b>Stan</b>	SHELLY.<nazwa>.light.0	0-100	Wartość wyjścia ściemnianego.
	SHELLY.<nazwa>.input.0	0-1	Stan wejścia binarnego.
	SHELLY.<nazwa>.longpush.0	0-1	Informacja o długim naciśnięciu przycisku.
	SHELLY.<nazwa>.light.0.power	dowolna liczba	Bieżący pobór mocy.
	SHELLY.<nazwa>.light.0.energy	dowolna liczba	Zużycie energii [W/m].
	SHELLY.<nazwa>.temperature	dowolna liczba	Wewnętrzna temperatura urządzenia.
	SHELLY.<nazwa>.overtemperature	0-1	Informacja o przegrzaniu urządzenia (wtedy wartość 1).
	SHELLY.<nazwa>.overload	0-1	Informacja o przeciążeniu wyjścia (wtedy wartość 1).
	SHELLY.<nazwa>.loaderror	0-1	Informacja o problemie z wyjściem ściemnianym (wtedy wartość 1).
	SHELLY.<nazwa>.light.0.overpower_value	dowolna liczba	Jeżeli jest przeciążenie przekaźnika, ten parametr zawiera informację o ile watów przekroczono limit.
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .
	E.SHELLY.<nazwa>.input_event.0	S/L	Informacja o zwarciu wejścia binarnego: S - krótkie L - długie <u>Tylko, gdy wejście w trybie Momentary</u>

## 12. Identyfikatory

### RGBW2 (tylko tryb White)

Czterokanałowy moduł do sterowania oświetleniem LED (12/24V). Każdy kanał może być sterowany niezależnie. Ponadto moduł wyposażony w pojedyncze wejście binarne.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.white.<0-3>	0-100	Ustawienie określonej jasności.
		on	Włącz.
		off	Wyłącz.
		toggle	Przełącz.
		<komen- da>;ramp:<rampa>	Wszystkie powyższe komendy z opcją z użycia rampy. Rampa określa czas przejścia z jednego stanu do innego. Maksymalna wartość rampy 5 [s].
<b>Stan</b>	SHELLY.<nazwa>.white.<0-3>	0-100	Wartość wyjścia ściemnianego.
	SHELLY.<nazwa>.input.0	0-1	Stan wejścia binarnego.
	SHELLY.<nazwa>.longpush.0	0-1	Informacja o długim naciśnięciu przycisku.
	SHELLY.<nazwa>.white.<0-3>.power	dowolna liczba	Bieżący pobór mocy
	SHELLY.<nazwa>.white.<0-3>.energy	dowolna liczba	Zużycie energii [W/m].
	SHELLY.<nazwa>.temperature	dowolna liczba	Wewnętrzna temperatura urządzenia.
	SHELLY.<nazwa>.overtemperature	0-1	Informacja o przegrzeniu urządzenia (wtedy wartość 1).
	SHELLY.<nazwa>.white.<0-3>.overpower	0-1	Informacja o przeciążeniu wyjścia (wtedy wartość 1).
	SHELLY.<nazwa>.loaderror	0-1	Informacja o problemie z wyjściem ściemnianym (wtedy wartość 1).
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .
	E.SHELLY.<nazwa>.input_event.0	S/L	Informacja o zwarciu wejścia binarnego: S - krótkie L - długie <u>Tylko, gdy wejście w trybie Momentary</u>

## Uni

Moduł z dwoma wejściami binarnymi oraz dwoma wyjściami bezpotencjałowymi.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.relay.<0-1>	0/off	Wyłącz.
		1/on	Włącz.
		toggle	Przełącz.
<b>Stan</b>	SHELLY.<nazwa>.relay.<0-1>	0-1	Stan wyjścia.
	SHELLY.<nazwa>.input.<0-1>	0-1	Stan wejścia binarnego.
	SHELLY.<nazwa>.longpush.<0-1>	0-1	Informacja o długim naciśnięciu przycisku.
	SHELLY.<nazwa>.adc.0	wg zakresu przetwornika	Wartość napięcia mierzona przez wbudowany przetwornik analogowo-cyfrowy.
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz Stan	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji Stan.
	E.SHELLY.<nazwa>.input_event.<0-1>	S/L	Informacja o zwarciu wejścia binarnego: S - krótkie L - długie Tylko, gdy wejście w trybie Momentary

## 12. Identyfikatory

### TRV

Zawór termostatyczny do sterowania ogrzewaniem.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.thermostat.0.schedule	0-1	Włącz/wyłącz sterowanie przez harmonogram.
	C.SHELLY.<nazwa>.thermostat.0.accelerated_heating	0-1	Włącz/wyłącz funkcję przyspieszonego ogrzewania.
	C.SHELLY.<nazwa>.thermostat.0.schedule_profile	1-5	Aktywowanie profilu harmonogramu ogrzewania o wybranym numerze.
	C.SHELLY.<nazwa>.thermostat.0.target	4-31	Ustawienie temperatury zadanej.
	C.SHELLY.<nazwa>.thermostat.0.ext_t	dowolna	Wysyłanie do głowicy termostatu wartości temperatury otoczenia mierzonej przez zewnętrzny czujnik.
	C.SHELLY.<nazwa>.thermostat.0.value_min_percent	0-100	Minimalna wartość otwarcia zaworu.
	C.SHELLY.<nazwa>.thermostat.0.boost_minutes	0-1440	Czas pracy w trybie przyspieszonego ogrzewania. Po jego upływie ogrzewanie wróci do trybu normalnego.
<b>Stan</b>	SHELLY.<nazwa>.thermostat.battery	0-100	Stan naładowania baterii.
	SHELLY.<nazwa>.thermostat.target	4-31	Wartość temperatury zadanej.
	SHELLY.<nazwa>.thermostat.enabled	0-1	Informacja o aktywacji automatycznego sterowania zaworem grzejnikowym.
	SHELLY.<nazwa>.thermostat.sensor	wg zakresu czujnika	Temperatura pomieszczenia.
	SHELLY.<nazwa>.thermostat.sensor.isvalid	0-1	Informacja, czy czujnik temperatury pracuje prawidłowo.
	SHELLY.<nazwa>.thermostat.pos	0-100	Pozycja zaworu grzejnikowego. -1 oznacza brak kalibracji zaworu.
	SHELLY.<nazwa>.thermostat.schedule	0-1	Informacja o aktywowaniu harmonogramu.
	SHELLY.<nazwa>.thermostat.profile	1-5	Informacja o numerze aktywowanego harmonogramu.
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .

## Vintage

### Ściemnialna żarówka LED

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.light.0	0-100	Ustawienie określonej jasności.
		on	Włącz.
		off	Wyłącz.
		toggle	Przełącz.
		<komen- da>;ramp:<rampa>	Wszystkie powyższe komendy z opcją użycia rampy. Rampa określa czas przejścia z jednego stanu do innego. Maksymalna wartość rampy 5 [s].
<b>Stan</b>	SHELLY.<nazwa>.light.0	0-100	Aktualna jasność.
	SHELLY.<nazwa>.light.0.power	dowolna liczba	Bieżący pobór mocy.
	SHELLY.<nazwa>.light.0.energy	dowolna liczba	Zużycie energii [W/m].
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz Stan	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .

## H&T

### Bezprzewodowy czujnik temperatury i wilgotności powietrza.

	Identyfikator	Wartość	Opis
<b>Stan</b>	SHELLY.<nazwa>.sensor.temperature	wg zakresu przetwornika	Mierzona temperatura.
	SHELLY.<nazwa>.sensor.humidity	0-100	Wilgotność względna.
	SHELLY.<nazwa>.sensor.battery	0-100	Stan naładowania baterii.
	SHELLY.<nazwa>.sensor.act_reasons	battery/button/periodic/poweron/sensor/alarm	Powód wzbudzenia czujnika.
	SHELLY.<nazwa>.sensor.ext_power	0-1	Informacja o podłączeniu do zewnętrznego źródła zasilania.
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .

## 12. Identyfikatory

### Smoke

Bezprzewodowy czujnik dymu.

	Identyfikator	Wartość	Opis
<b>Stan</b>	SHELLY.<nazwa>.sensor.temperature	wg zakresu przetwornika	Mierzona temperatura.
	SHELLY.<nazwa>.sensor.smoke	0-1	Stan wykrycia dymu.
	SHELLY.<nazwa>.sensor.battery	0-100	Stan naładowania baterii.
<b>Zdarzenie</b>	<b>E</b> .SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .

### Flood

Bezprzewodowy czujnik zalania.

	Identyfikator	Wartość	Opis
<b>Stan</b>	SHELLY.<nazwa>.sensor.temperature	wg zakresu przetwornika	Mierzona temperatura.
	SHELLY.<nazwa>.sensor.flood	0-1	Stan wykrycia zalania.
	SHELLY.<nazwa>.sensor.battery	0-100	Stan naładowania baterii.
	SHELLY.<nazwa>.sensor.act_reasons	battery/button/periodic/poweron/sensor/alarm	Powód wzbudzenia czujnika.
<b>Zdarzenie</b>	<b>E</b> .SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b>

### Door/Window 1/2

Bezprzewodowy czujnik otwarcia drzwi/okna z pomiarem nasłonecznienia i detekcją ruchu (pomiar wibracji).

	Identyfikator	Wartość	Opis
<b>Stan</b>	SHELLY.<nazwa>.sensor.state	0-1	Stan otwarcia okna/drzwi.
	SHELLY.<nazwa>.sensor.tilt	0-180	Kąt otwarcia okna/drzwi.
	SHELLY.<nazwa>.sensor.vibration	0-1	Informacja o wykryciu wibracji.
	SHELLY.<nazwa>.sensor.act_reasons	battery/button/periodic/poweron/sensor/movement/temperature/light	Powód wzbudzenia czujnika.
	SHELLY.<nazwa>.sensor.lux	wg zakresu przetwornika	Poziom jasności w luxach.
	SHELLY.<nazwa>.sensor.battery	0-100	Stan naładowania baterii.
	SHELLY.<nazwa>.sensor.temperature	wg zakresu przetwornika	Mierzona temperatura.
<b>Zdarzenie</b>	<b>E</b> .SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .



## Motion

Bezprzewodowy czujnik ruchu z pomiarem nasłonecznienia.

	Identyfikator	Wartość	Opis
<b>Stan</b>	SHELLY.<nazwa>.motion	0-1	Informacja o wykryciu ruchu.
	SHELLY.<nazwa>.active	0-1	Informacja, czy detekcja ruchu jest aktywowana.
	SHELLY.<nazwa>.vibration	0-1	Informacja o wykryciu wibracji.
	SHELLY.<nazwa>.lux	wg zakresu przetwornika	Poziom jasności w luxach.
	SHELLY.<nazwa>.battery	0-100	Stan naładowania baterii.
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .

## Sense

Bezprzewodowe urządzenie pomiarowe wyposażone w czujniki: temperatury, wilgotności, nasłonecznienia, detektor ruchu. Ponadto umożliwia emitowanie kodów IR do dowolnych urządzeń multimedialnych. Emisja kodów IR nie jest dostępna poprzez integrację protokołem MQTT, natomiast dostępna jest przez aplikację Shelly, API HTTP lub interfejs konfiguracyjny urządzenia.

	Identyfikator	Wartość	Opis
<b>Stan</b>	SHELLY.<nazwa>.sensor.motion	0-1	Informacja o wykryciu ruchu.
	SHELLY.<nazwa>.sensor.charger	0-1	Informacja o podłączeniu do zewnętrznego źródła zasilania.
	SHELLY.<nazwa>.sensor.temperature	wg zakresu przetwornika	Temperatura pomieszczenia.
	SHELLY.<nazwa>.sensor.lux	wg zakresu przetwornika	Poziom jasności w luxach.
	SHELLY.<nazwa>.sensor.battery	0-100	Stan naładowania baterii.
	SHELLY.<nazwa>.sensor.humidity	0-100	Wilgotność względna [%].
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .

## 12. Identyfikatory

### Button1

Bezprzewodowy przycisk rozróżniający 4 rodzaje naciśnięć: krótkie, 2x krótkie, 3x krótkie, długie. Czas reakcji przycisku w trybie bateryjnym wynosi około 2s oraz około 100ms w trybie zasilania przez USB.

	Identyfikator	Wartość	Opis
Stan	SHELLY.<nazwa>.sensor.battery	0-100	Stan naładowania baterii.
	SHELLY.<nazwa>.sensor.charger	0-1	Informacja o podłączeniu do zewnętrznego źródła zasilania.
Zdarzenie	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .
	E.SHELLY.<nazwa>.input_event.0	S/SS/SSS/L	Informacja o naciśnięciu przycisku: S - krótkie SS- 2x krótkie SSS - 3x krótkie L - długie

### i3

Trzykanałowy moduł wejść binarnych z detekcją różnych typów naciśnięcia przycisków.

	Identyfikator	Wartość	Opis
Stan	SHELLY.<nazwa>.input.<0-2>	0-1	Stan wejścia.
	SHELLY.<nazwa>.temperature_status	normal/ high/ veryhigh	Status temperatury urządzenia.
Zdarzenie	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .
	E.SHELLY.<nazwa>.input_event.<0-2>	S/L/SS/SSS/ SL/LS	Informacja o naciśnięciu przycisku: S - krótkie SS- 2x krótkie SSS - 3x krótkie L - długie SL - sekwencja krótko-długo LS - sekwencja długo-krótko

## Gas

Detektor gazu LPG lub CNG.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.valve.0	open	Otwórz zawór.
		close	Zamknij zawór.
	C.SHELLY.<nazwa>.sensor.0	mute	Wyciszenie alarmu
		unmute	Anulowanie wyciszenia alarmu.
		test	Uruchomienie procedury testującej detektor.
<b>Stan</b>	SHELLY.<nazwa>.sensor.operation	unknown/warmup/ normal/fault	Stan detektora: unknown - Stan nieustalony. warmup - Detektor się rozgrzewa. normal - Normalna praca detektora. fault - Błąd detektora.
	SHELLY.<nazwa>.sensor.gas	unknown/none/ mild/heavy/test	Stan alarmu detektora: unknown - Stan alarmu nieustalony. none - Brak alarmu. mild - Delikatny wyciek gazu. heavy - Poważny wyciek gazu. test - Stan po przeprowadzeniu procedury testowej.
	SHELLY.<nazwa>.sensor.self_test	not_completed/ completed/run- ning/pending	Stan procedury testowej: not_completed - Procedura testowa niedokończona. completed - Procedura testowa zakończona. running - Procedura w trakcie. pending - Procedura oczekuje na rozpoczęcie.
	SHELLY.<nazwa>.sen- sor.concentration	0-65535	Koncentracja gazu [cząłka/m3], -1 gdy błąd.
	SHELLY.<nazwa>.valve.state	unknown/closed/ opened/not_con- nected/failure/ closing/opening/ checking	Stan zaworu gazu: unknown - Stan nieznaný. closed - Zawór zamknięty. opened - Zawór otwarty. not_connected - Zawór nie jest podłączony. failure - Wykryto awarię zaworu. closing - Zawór jest w trakcie zamykania. opening - Zawór jest w trakcie otwierania. checking - Stan zaworu jest w trakcie sprawdzenia.
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identy- fikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .

## 12. Identyfikatory

### EM

Dwukanałowy licznik energii. Moduł wyposażony jest także w pojedyncze wyjście niskoprądowe.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.relay.0	on	Włącz wyjście.
		off	Wyłącz wyjście.
		toggle	Przełącz wyjście.
	C.SHELLY.<nazwa>.emeter.0	reset	Reset wszystkich danych licznika.
		clear_t	Skasowanie licznika energii pobranej.
		clear_r	Skasowanie licznika oddanej.
<b>Stan</b>	SHELLY.<nazwa>.emeter.0.energy	dowolna liczba	Licznik energii pobranej [W/min].
	SHELLY.<nazwa>.emeter.0.returned_energy	dowolna liczba	Licznik energii oddanej [W/min].
	SHELLY.<nazwa>.emeter.0.total	dowolna liczba	Całkowita wartość energii pobranej [Wh].
	SHELLY.<nazwa>.emeter.0.total_returned	dowolna liczba	Całkowita wartość energii oddanej [Wh].
	SHELLY.<nazwa>.emeter.0.power	dowolna liczba	Moc czynna [W].
	SHELLY.<nazwa>.emeter.0.reactive_power	dowolna liczba	Moc bierna [W].
	SHELLY.<nazwa>.emeter.0.voltage	dowolna liczba	Napięcie [V].
	SHELLY.<nazwa>.relay.0	0-1	Stan wyjścia.
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz Stan	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .

## 3EM

Trójfazowy licznik energii elektrycznej.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.relay.0	on	Włącz wyjście.
		off	Wyłącz wyjście.
		toggle	Przełącz wyjście.
	C.SHELLY.<nazwa>.emeter.<0-2>	reset	Reset wszystkich danych licznika.
		clear_t	Skasowanie licznika energii pobranej.
		clear_r	Skasowanie licznika oddanej.
<b>Stan</b>	SHELLY.<nazwa>.emeter.<0-2>.energy	dowolna liczba	Licznik energii pobranej [W/min].
	SHELLY.<nazwa>.emeter.<0-2>.returned_energy	dowolna liczba	Licznik energii oddanej [W/min].
	SHELLY.<nazwa>.emeter.<0-2>.total	dowolna liczba	Całkowita wartość energii pobranej [Wh].
	SHELLY.<nazwa>.emeter.<0-2>.total_returned	dowolna liczba	Całkowita wartość energii oddanej [Wh].
	SHELLY.<nazwa>.emeter.<0-2>.power	dowolna liczba	Moc czynna [W].
	SHELLY.<nazwa>.emeter.<0-2>.reactive_power	dowolna liczba	Moc bierna [W].
	SHELLY.<nazwa>.emeter.<0-2>.voltage	dowolna liczba	Napięcie [V].
	SHELLY.<nazwa>.emeter.<0-2>.pf	0-1	Współczynnik mocy.
	SHELLY.<nazwa>.relay.0	0-1	Stan wyjścia.
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz Stan	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .

## 12. Identyfikatory

### Generacja 2

#### Seria "Plus"

Moduły w serii **Plus** to moduły dopuszkowe drugiej generacji.

#### Plus 1/1 PM

Moduł dopuszkowy wyposażony w pojedynczy przekaźnik oraz wejście binarne. Moduł 1 PM dodatkowo wyposażony jest licznik energii.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.relay.0	0/off	Wyłącz.
		1/on	Włącz.
		toggle	Przełącz.
		on/off/ 1/0;flip:xx	Wykonuje zmianę stanu przekaźnika na określony interwał czasu. Przykłady pod tabelą. xx - czas trwania komendy, może być wartość ułamkowa.
<b>Stan</b>	SHELLY.<nazwa>.relay.0	0-1	Stan przekaźnika.
	SHELLY.<nazwa>.input.0	0-1	Stan wejścia binarnego.
	SHELLY.<nazwa>.relay.0.errors	overtemp/ overpower/ overvoltage	Informacja o błędach modułu.
	SHELLY.<nazwa>.relay.0.power (tylko wersja PM)	dowolna liczba	Pobór mocy.
	SHELLY.<nazwa>.relay.0.energy (tylko wersja PM)	dowolna liczba	Zużycie energii [W/h].
	SHELLY.<nazwa>.relay.0.temperatu- re (tylko wersja PM)	dowolna liczba	Wewnętrzna temperatura urządzenia.
	SHELLY.<nazwa>.relay.0.voltage (tylko wersja PM)	dowolna liczba	Mierzone napięcie [V].
	SHELLY.<nazwa>.relay.0.current (tylko wersja PM)	dowolna liczba	Mierzony prąd [A].
	SHELLY.<nazwa>.relay.0.pf (tylko wersja PM)	0-1	Współczynnik mocy.
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identy- fikatora>	Patrz Stan	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .
	E.SHELLY.<nazwa>.input_event.0	S/SS/L	Informacja o zwarceniu wejścia binarnego: S - krótkie SS - 2x krótkie L - długie <u>Tylko, gdy wejście w trybie Button</u>
	E.SHELLY.<nazwa>.longpush.0	0-1	Zdarzenie dla długiego zwarcia wejścia. <u>Tylko, gdy wejście w trybie Button</u>



- C.SHELLY.test.relay.0=on;flip:0.1  
Przełącznik zostanie włączony na 100ms.
- C.SHELLY.test.relay.0=0;flip:2  
Przełącznik zostanie wyłączony na 2 sekundy.

## Plus 2 PM

Moduł wyposażony w dwa przekaźniki z pomiarem mocy i dwa wejścia binarne. Może pracować w trybie roletowym lub jako dwa niezależne przekaźniki.

### Tryb roletowy

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.roller.0	up	W górę
		down	W dół.
		stop	Stop.
		rc	Kalibracja rolety.
		0-100	Ustawienie rolety na określoną pozycję. 100 - pełne otwarcie 0 - pełne zamknięcie
		rel:x	Zmiana położenia rolety względem aktualnej pozycji. x - zmiana wyrażona w % w zakresie -100..100 Komenda korzysta z formuły: $pozycja\_zadana = pozycja\_aktualna + x$ Jeżeli pozycja zadana wykracza po zakres, wówczas przyjmowana jest wartość maksymalna lub minimalna.
		up;duration:x	Ruch rolety w górę przez określony czas x
down;duration:x	Ruch rolety w dół przez określony czas x		

## 12. Identyfikatory

<b>Stan</b>	SHELLY.<nazwa>.roller.0	up/down/stop	Stan ruchu rolety.
	SHELLY.<nazwa>.roller.0.pos	0-100	Aktualna pozycja. Opcja dostępna tylko po kalibracji. Wartość -1 oznacza błędną pozycję (brak kalibracji).
	SHELLY.<nazwa>.roller.0.target	0-100	Zadana pozycja rolety.
	SHELLY.<nazwa>.roller.0.state	open/closed/opening/closing/stopped/calibrating	Informacja o stanie: open - pełne otwarcie; closed - pełne zamknięcie; opening - w trakcie otwierania; closing - w trakcie zamykania; stopped - ruch zatrzymany w pozycji pomiędzy pełnym otwarciem a zamknięciem; calibrating - w trakcie kalibracji;
	SHELLY.<nazwa>.roller.0.power	dowolna liczba	Bieżący pobór mocy.
	SHELLY.<nazwa>.roller.0.energy	dowolna liczba	Zużycie energii [W/h].
	SHELLY.<nazwa>.roller.0.temperature	dowolna liczba	Wewnętrzna temperatura urządzenia.
	SHELLY.<nazwa>.roller.0.voltage	dowolna liczba	Mierzone napięcie [V].
	SHELLY.<nazwa>.roller.0.pf	0-1	Współczynnik mocy.
	SHELLY.<nazwa>.roller.0.current	dowolna liczba	Mierzony prąd [A].
	SHELLY.<nazwa>.roller.0.positioning	0-1	0 - gdy silnik nieskalibrowany; 1 - gdy silnik skalibrowany i pozycjonowanie jest możliwe;
SHELLY.<nazwa>.roller.0.errors	wg dokumentacji urządzenia	Informacja o błędach modułu.	
SHELLY.<nazwa>.input.<0-1>	0-1	Stan wejścia binarnego.	
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz Stan	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .
	E.SHELLY.<nazwa>.longpush.0	0-1	Zdarzenie dla długiego zwarcia wejścia. <u>Tylko, gdy wejście w trybie Button</u>
	E.SHELLY.<nazwa>.input_event.0	S/SS/L	Informacja o zwarcu wejścia binarnego: S - krótkie SS - 2x krótkie L - długie <u>Tylko, gdy wejście w trybie Button</u>



## Tryb przekaźnikowy

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.relay.<0-1>	0/off	Wyłącz.
		1/on	Włącz.
		toggle	Przełącz.
		on/off/ 1/0;flip:xx	Wykonuje zmianę stanu przekaźnika na określony interwał czasu. xx - czas trwania komendy, może być wartość ułamkowa. Przykłady pod tabelą.
<b>Stan</b>	SHELLY.<nazwa>.relay.<0-1>	0-1	Stan przekaźnika.
	SHELLY.<nazwa>.input.<0-1>	0-1	Stan wejścia binarnego.
	SHELLY.<nazwa>.relay.<0-1>.errors	overtemp/ overpower/ overvoltage	Informacja o błędach modułu.
	SHELLY.<nazwa>.relay.<0-1>.power	dowolna liczba	Pobór mocy.
	SHELLY.<nazwa>.relay.<0-1>.energy	dowolna liczba	Zużycie energii [W/h].
	SHELLY.<nazwa>.relay.<0-1>.temperature	dowolna liczba	Wewnętrzna temperatura urządzenia.
	SHELLY.<nazwa>.relay.<0-1>.voltage	dowolna liczba	Mierzone napięcie [V].
	SHELLY.<nazwa>.relay.<0-1>.current	dowolna liczba	Mierzony prąd [A].
	SHELLY.<nazwa>.relay.<0-1>.pf	0-1	Współczynnik mocy.
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz Stan	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .
	E.SHELLY.<nazwa>.input_event.<0-1>	S/SS/L	Informacja o zwarceniu wejścia binarnego: S - krótkie SS - 2x krótkie L - długie <u>Tylko, gdy wejście w trybie Button</u>
	E.SHELLY.<nazwa>.longpush.0	0-1	Zdarzenie dla długiego zwarcia wejścia. <u>Tylko, gdy wejście w trybie Button</u>



- C.SHELLY.test.relay.0=on;flip:0.1  
Przełącznik zostanie włączony na 100ms.
- C.SHELLY.test.relay.0=0;flip:2  
Przełącznik zostanie wyłączony na 2 sekundy.

## 12. Identyfikatory

### I4

Moduł czterech wejść binarnych.

	Identyfikator	Wartość	Opis
<b>Stan</b>	SHELLY.<nazwa>.input.<0-3>	0-1	Stan wejścia binarnego.
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz <b>Stan</b>	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .
	E.SHELLY.<nazwa>.input_event.<0-3>	S/SS/L	Informacja o zwarciu wejścia binarnego: S - krótkie SS - 2x krótkie L - długie <u>Tylko, gdy wejście w trybie Button</u>
	E.SHELLY.<nazwa>.longpush.<0-3>	0-1	Zdarzenie dla długiego zwarcia wejścia. <u>Tylko, gdy wejście w trybie Button</u>

## Seria modułów "Pro"

Moduły z serii Pro przeznaczone są do montażu w szafach rozdzielczych. Wyposażone są w łączność bezprzewodową WiFi oraz złącze Ethernet.

### Pro 1/ Pro 1PM

Moduł rozdzielnicowy wyposażony w jeden przekaźnik oraz dwa wejścia binarne. Wersja PM ponadto wyposażona w licznik energii elektrycznej.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.relay.0	0/off	Wyłącz.
		1/on	Włącz.
		toggle	Przełącz.
		on/off/ 1/0;flip:xx	Wykonuje zmianę stanu przekaźnika na określony interwał czasu. Przykłady pod tabelą. xx - czas trwania komendy, może być wartość ułamkowa.
<b>Stan</b>	SHELLY.<nazwa>.relay.0	0-1	Stan przekaźnika.
	SHELLY.<nazwa>.input.<0-1>	0-1	Stan wejścia binarnego.
	SHELLY.<nazwa>.relay.0.errors	overtemp/ overpower/ overvoltage	Informacja o błędach modułu.
	SHELLY.<nazwa>.relay.0.power (tylko wersja PM)	dowolna liczba	Pobór mocy.
	SHELLY.<nazwa>.relay.0.energy (tylko wersja PM)	dowolna liczba	Zużycie energii [W/h].
	SHELLY.<nazwa>.relay.0.temperature (tylko wersja PM)	dowolna liczba	Wewnętrzna temperatura urządzenia.
	SHELLY.<nazwa>.relay.0.voltage (tylko wersja PM)	dowolna liczba	Mierzone napięcie [V].
	SHELLY.<nazwa>.relay.0.current (tylko wersja PM)	dowolna liczba	Mierzony prąd [A].
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfikatora>	Patrz Stan	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .
	E.SHELLY.<nazwa>.input_event.<0-1>	S/SS/L	Informacja o zwarcie wejścia binarnego: S - krótkie SS - 2x krótkie L - długie <u>Tylko, gdy wejście w trybie Button</u>
	E.SHELLY.<nazwa>.longpush.<0-1>	0-1	Zdarzenie dla długiego zwarcia wejścia. <u>Tylko, gdy wejście w trybie Button</u>



- C.SHELLY.test.relay.0=on;flip:0.1  
Przekaźnik zostanie włączony na 100ms.
- C.SHELLY.test.relay.0=0;flip:2  
Przekaźnik zostanie wyłączony na 2 sekundy.

## 12. Identyfikatory

### Pro 2

Moduł rozdzielnicowy wyposażony w dwa przekaźniki i dwa wejścia binarne.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.SHELLY.<nazwa>.relay.<0-1>	0/off	Wyłącz.
		1/on	Włącz.
		toggle	Przełącz.
		on/off/ 1/0;flip:xx	Wykonuje zmianę stanu przekaźnika na określony interwał czasu. Przykłady pod tabelą. xx - czas trwania komendy, może być wartość ułamkowa.
<b>Stan</b>	SHELLY.<nazwa>.relay.<0-1>	0-1	Stan przekaźnika.
	SHELLY.<nazwa>.input.<0-1>	0-1	Stan wejścia binarnego.
	SHELLY.<nazwa>.relay.<0-1>.errors	overtemp/ overpower/ overvoltage	Informacja o błędach modułu.
	SHELLY.<nazwa>.relay.<0-1>.power (tylko wersja PM)	dowolna liczba	Pobór mocy.
	SHELLY.<nazwa>.relay.<0-1>.energy (tylko wersja PM)	dowolna liczba	Zużycie energii [W/h].
	SHELLY.<nazwa>.relay.<0-1>.temperatu- re (tylko wersja PM)	dowolna liczba	Wewnętrzna temperatura urządzenia.
	SHELLY.<nazwa>.relay.<0-1>.voltage (tylko wersja PM)	dowolna liczba	Mierzone napięcie [V].
	SHELLY.<nazwa>.relay.<0-1>.current (tylko wersja PM)	dowolna liczba	Mierzony prąd [A].
SHELLY.<nazwa>.relay.<0-1>.pf (tylko wersja PM)	0-1	Współczynnik mocy.	
<b>Zdarzenie</b>	E.SHELLY.<nazwa>.<reszta_identyfika- tora>	Patrz Stan	Moduł Base rozgłasza zdarzenia dla zmian wszystkich parametrów z sekcji <b>Stan</b> .
	E.SHELLY.<nazwa>.input_event.<0-1>	S/SS/L	Informacja o zwarceniu wejścia binarnego: S - krótkie SS - 2x krótkie L - długie <u>Tylko, gdy wejście w trybie Button</u>
	E.SHELLY.<nazwa>.longpush.<0-1>	0-1	Zdarzenie dla długiego zwarcia wejścia. <u>Tylko, gdy wejście w trybie Button</u>

## Pro 2 PM

Moduł rozdzielnicowy wyposażony w dwa przekaźniki z pomiarem mocy i dwa wejścia binarne. Może pracować w trybie roletowym lub jako dwa niezależne przekaźniki.

Opis komend, stanu i zdarzeń jest identyczny jak dla modułu Plus 2PM.

## Pro 4 PM

Moduł rozdzielnicowy wyposażony w cztery przekaźniki i cztery wejścia binarne. Każdy przekaźnik ma także licznik energii elektrycznej.

Opis komend, stanu i zdarzeń jest identyczny jak dla modułu Plua 1/1PM z tą różnicą, że zakres numeracji przekaźników i wejść jest od 0-3.

## 12.9. ULUX

Identyfikatory ULUX umożliwiają zarządzanie włącznikami u::Lux. Poniższa tabela zawiera przegląd wszystkich identyfikatorów ULUX i ich główne cechy. Szczegółowy opis poszczególnych identyfikatorów wraz z przykładami zastosowania znajduje się w dalszej części podrozdziału.

Identyfikator	Skrócony opis	Stan	Zdarzenia	Komenda
ULUX.key	Zdarzenie informujące o naciśnięciu klawisza.	-	+	+
ULUX.sound	Komenda umożliwiająca odtworzenie dźwięku.	-	-	+
ULUX.screensaver	Komenda sterująca wygaszaczem ekranu.	-	-	+
ULUX.page	Komenda zmieniająca aktualnie wyświetlaną stronę.	-	-	+
ULUX.led	Komenda sterująca diodą LED klawisza.	-	-	+
ULUX.text	Komendą umożliwiającą wyświetlenie dowolnego tekstu na wyświetlaczu włącznika.	-	-	+
ULUX.volume	Identyfikator sterujący głośnością dźwięku włącznika oraz informujący o bieżącym stanie i zmianach nastaw.	+	+	+
ULUX.brightness	Identyfikator informujący o jasności w pomieszczeniu oraz o jej zmianach.	+	+	-
ULUX.motion	Identyfikator informujący o wykryciu ruchu przez czujkę.	+	+	-
ULUX.temperature	Identyfikator informujący o temperaturze w pomieszczeniu oraz o jej zmianach.	+	+	-
ULUX.humidity	Identyfikator informujący o wilgotności powietrza w pomieszczeniu oraz o jej zmianach.	+	+	-
ULUX.CO2	Identyfikator informujący o stężeniu dwutlenku węgla w pomieszczeniu oraz o jego zmianach.	+	+	-
ULUX.input1	Identyfikator informujący o stanie wejścia binarnego nr 1 oraz jego zmianach.	+	+	-
ULUX.input2	Identyfikator informujący o stanie wejścia binarnego nr 2 oraz jego zmianach.	+	+	-

## ULUX.key

Zdarzenie informujące o naciśnięciu klawisza. Zdarzenie jest wywoływane tylko wtedy, gdy aktor pracuje w trybie **wartość rzeczywista**. Funkcjonalność modułu Base umożliwia rozróżnianie krótkiego i długiego naciśnięcia klawisza.

	Identyfikator	Wartość	Opis
<b>Zdarzenie</b>	E.ULUX.key.ID.A ID – ID włącznika u::Lux A – numer aktora	hit/make/break	Zdarzenie informujące o naciśnięciu lub zwolnieniu klawisza.



### 1. Zdarzenie informujące o naciśnięciu przycisku

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Krótkie naciśnięcie

**Kanał:** E.ULUX.key.10.1

**Dane:** hit

Odebrano zdarzenie z włącznika o identyfikatorze nr 10 o krótkim naciśnięciu klawisza przypisanego do aktora nr 1.

- Długie naciśnięcie

**Kanał:** E.ULUX.key.10.1

**Dane:** make

Odebrano zdarzenie z włącznika o identyfikatorze nr 10 o długim naciśnięciu klawisza przypisanego do aktora nr 1.

- Puszczanie przycisku

**Kanał:** E.ULUX.key.10.1

**Dane:** break

Odebrano zdarzenie z włącznika o identyfikatorze nr 10 o długim puszczeniu klawisza przypisanego do aktora nr 1.

## 12. Identyfikatory

### ULUX.sound

Komenda ta umożliwia odtworzenie dowolnego dźwięku wgranego do włącznika u::Lux.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.ULUX.sound.ID.S ID – ID włącznika u::Lux S – numer pliku dźwiękowego w konfiguracji u::Lux Config	0-100	Komenda odtwarzająca dźwięk we włączniku u::Lux z określoną głośnością.



#### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Odtworzenie dźwięku w pojedynczym włączniku**

**Kanał:** C.ULUX.sound.17.1

**Dane:** 100

Odtwórz dźwięk nr 1 we włączniku o identyfikatorze 17 z 100% głośnością.

- **Odtworzenie dźwięku we wszystkich włącznikach**

**Kanał:** C.ULUX.sound.all.1

**Dane:** 40

Odtwórz dźwięk nr 1 we wszystkich włącznikach u::Lux dostępnych w instalacji, z głośnością 40%.



## ULUX.screensaver

Komenda sterująca wygaszaczem ekranu na jednym, bądź wszystkich włącznikach u::Lux.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.ULUX.screensaver.ID ID – ID włącznika u::Lux	on/off	Komenda włączająca/wyłączająca wygaszacz ekranu.



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Kanał:** C.ULUX.screensaver.1  
**Dane:** on  
Włącz wygaszacz ekranu we włączniku z identyfikatorem 1.
- **Kanał:** C.ULUX.screensaver.1  
**Dane:** off  
Wyłącz wygaszacz ekranu we włączniku z identyfikatorem 1.
- **Kanał:** C.ULUX.screensaver.all  
**Dane:** on  
Włącz wygaszacz ekranu we wszystkich włącznikach.
- **Kanał:** C.ULUX.screensaver.all  
**Dane:** off  
Wyłącz wygaszacz ekranu we wszystkich włącznikach.

## ULUX.page

Komenda przełączająca aktualnie wyświetlaną stronę. Komenda może być wysłana do jednego lub wszystkich włączników.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.ULUX.page.ID ID – ID włącznika u::Lux	1-32	Komenda przełączająca stronę aktualnie wyświetlaną na ekranie.



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Kanał:** C.ULUX.page.1  
**Dane:** 5  
Wyświetl stronę nr 5 we włączniku z identyfikatorem 1.
- **Kanał:** C.ULUX.page.all  
**Dane:** 5  
Wyświetl stronę nr 5 we wszystkich włącznikach.

## 12. Identyfikatory

### ULUX.led

Komenda sterująca diodą LED przypisaną do danego aktora. Umożliwia zmianę zarówno koloru diody LED, jak i sposobu jej świecenia. Wysłanie tej komendy zmienia stan diody LED we wszystkich włącznikach, które posiadają odniesienie do aktora o wskazanym numerze. Aby działanie komendy było widoczne, na ekranie włącznika musi być wyświetlona strona, na której jest klawisz odnoszący się do danego aktora. Efekt działania komendy będzie widoczny do czasu odebrania następczej komendy typu ULUX.led wysłanej do tego samego aktora lub momentu restartu włącznika.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.ULUX.led.A A – Numer aktora	tryb:kolor	Komenda ustawiająca kolor diody LED oraz tryb jej świecenia.

Dostępne kolory:

- *off* – dioda wyłączona;
- *red* – czerwony;
- *green* – zielony;
- *yellow* – żółty;
- *blue* – niebieski;
- *magenta* – magenta;
- *cyan* – cyjan;
- *white* – biały;

Tryby świecenia:

- *0* – dioda stale włączona;
- *1* – miganie co sekundę;
- *2* – miganie co pół sekundy;
- *3* – miganie co ćwierć sekundy;
- *4* – miganie asymetryczne: dioda włączona przez pół sekundy, a przez 1/4 sekundy wyłączona;
- *5* – miganie asymetryczne: dioda włączona przez 1/2 sekundy, wyłączona przez 4 sekundy;
- *6* – miganie asymetryczne: dioda włączona przez 1/2 sekundy, wyłączona przez 16 sekund;
- *7* – miganie asymetryczne: dioda włączona przez 1/2 sekundy, wyłączona przez 32 sekundy;



#### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Kanał:** C.ULUX.led.1  
**Dane:** 0:off  
Wyłącz diodę przypisaną do aktora nr 1.
- **Kanał:** C.ULUX.led.1  
**Dane:** 0:red  
Ustaw kolor czerwony i świecenie stałe diody LED przypisanej do aktora nr 1.
- **Kanał:** C.ULUX.led.1  
**Dane:** 1:green  
Ustaw kolor zielony i miganie co sekundę diody LED przypisanej do aktora nr 1.

## ULUX.text

Komenda umożliwiająca wyświetlenie dowolnego tekstu na wyświetlaczu jednego lub wszystkich włączników. Uwaga! Wyświetlacz nie obsługuje polskich znaków.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.ULUX.text.ID.T <b>ID</b> – ID włącznika u::Lux <b>T</b> – identyfikator tekstu (zgodnie z konfiguracją w uLux::Config).	treść komunikatu	Komenda ustawiająca wyświetlenie dowolnego tekstu na ekranie.



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- Kanał:** C.ULUX.text.10.1  
**Dane:** Przykładowy tekst  
 Wyświetl tekst „Przykładowy tekst” w zmiennej tekstowej nr 1 na ekranie włącznika o identyfikatorze 10.
- Kanał:** C.ULUX.text.all.1  
**Dane:** Przykładowy tekst  
 Wyświetl tekst „Przykładowy tekst” w zmiennej tekstowej nr 1 na ekranie wszystkich włączników.

## ULUX.volume

Identyfikator przeznaczony do sterowania i wyświetlania bieżącej głośności włącznika.

	Identyfikator	Wartość	Opis
<b>Stan</b>	ULUX.volume.ID <b>ID</b> – ID włącznika u::Lux	0-100	Aktualna głośność włącznika.
<b>Zdarzenie</b>	E.ULUX.volume.ID	0-100	Zdarzenie informujące o zmianie głośności włącznika.
<b>Komenda</b>	C.ULUX.volume.ID	0-100	Komenda ustawiająca głośność dźwięków odtwarzanych przez włącznik.



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- Nazwa:** C.ULUX.volume.10  
**Wartość:** 60  
 Ustaw głośność włącznika o identyfikatorze nr 10 na 60%.
- Nazwa:** C.ULUX.volume.all  
**Wartość:** 60  
 Ustaw głośność wszystkich włączników na 60%.

### 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- Kanał:** E.ULUX.volume.10  
**Dane:** 60  
 Ustawiono głośność włącznika o identyfikatorze nr 10 na 60%.

## 12. Identyfikatory

### ULUX.brightness

Identyfikator informujący o bieżącej wartości jasności zmierzonej przez wbudowany czujnik oświetlenia oraz o zmianach tej wartości. Wartość wyrażana w luksach.

	Identyfikator	Wartość	Opis
<b>Stan</b>	ULUX.brightness.ID	0-65535	Aktualna jasność zmierzona przez czujnik jasności.
	ID – ID włącznika u::Lux		
<b>Zdarzenie</b>	E.ULUX.brightness.ID	0-65535	Zdarzenie informujące o zmianie jasności.



#### 1. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- **Kanał:** E.ULUX.brightness.1

**Dane:** 500

Jasność zmierzona przez czujnik oświetlenia we włączniku o identyfikatorze 1 osiągnęła wartość 500 luksów.

Opisane dalej identyfikatory występują tylko, jeżeli zostały podłączone moduły rozszerzeń do włączników **u::Lux**. Jeżeli w Twojej instalacji nie ma modułów rozszerzeń możesz pominąć dalszą część tego rozdziału.

### ULUX.motion

Identyfikator informujący o wykryciu ruchu przez czujkę oraz przechowujący jej bieżący stan.

	Identyfikator	Wartość	Opis
<b>Stan</b>	ULUX.motion.ID	0-1	Aktualny stan czujki ruchu.
	ID – ID włącznika u::Lux		
<b>Zdarzenie</b>	E.ULUX.motion.ID	0-1	Zdarzenie informujące o wykryciu ruchu.



#### 1. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- **Kanał:** E.ULUX.motion.1

**Dane:** 1

Czujka ruchu podłączona do włącznika z identyfikatorem nr 1 wykryła ruch.

## ULUX.temperature

Identyfikator informujący o bieżącej wartości temperatury powietrza zmierzonej przez moduł dodatkowy oraz o zmianach tej wartości.

	Identyfikator	Wartość	Opis
<b>Stan</b>	ULUX.temperature.ID ID – ID włącznika u::Lux	-55°C do +55°C	Aktualna temperatura powietrza zmierzona przez czujnik.
<b>Zdarzenie</b>	E.ULUX.temperature.ID	-55°C do +55°C	Zdarzenie informujące o zmianie temperatury



### 1. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- **Kanał:** E.ULUX.temperature.1

**Dane:** 24.5

Temperatura zmierzona przez czujnik temperatury włącznika o identyfikatorze 1 osiągnęła wartość 24.5 stopni Celsjusza.

## ULUX.humidity

Identyfikator informujący o bieżącej wartości wilgotności powietrza zmierzonej przez moduł dodatkowy oraz o zmianach tej wartości.

	Identyfikator	Wartość	Opis
<b>Stan</b>	ULUX.humidity.ID ID – ID włącznika u::Lux	0-100%	Aktualna wilgotność powietrza zmierzona przez czujnik.
<b>Zdarzenie</b>	E.ULUX.humidity.ID	0-100%	Zdarzenie informujące o zmianie wilgotności powietrza.



### 1. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- **Kanał:** E.ULUX.humidity.1

**Dane:** 55

Wilgotność zmierzona przez czujnik wilgotności włącznika o identyfikatorze 1 osiągnęła wartość 55%.

## 12. Identyfikatory

### ULUX.CO2

Identyfikator informujący o bieżącym stężeniu dwutlenku węgla w powietrzu zmierzonym przez moduł dodatkowy oraz o zmianach tej wartości.

	Identyfikator	Wartość	Opis
<b>Stan</b>	ULUX.CO2.ID ID – ID włącznika u::Lux	400ppm – 4000ppm	Aktualne stężenie dwutlenku węgla zmierzone przez czujnik.
<b>Zdarzenie</b>	E.ULUX.CO2.ID	400ppm – 4000ppm	Zdarzenie informujące o zmianie zawartości dwutlenku węgla w powietrzu.



#### 1. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- **Kanał:** E.ULUX.CO2.1

**Dane:** 1200

Stężenie dwutlenku węgla w powietrzu zmierzone przez czujnik podłączony do włącznika o identyfikatorze nr 1 osiągnęło wartość 1200ppm.

### ULUX.input1

Identyfikator informujący o aktualnym stanie wejścia binarnego nr 1 oraz zmianach jego stanu.

	Identyfikator	Wartość	Opis
<b>Stan</b>	ULUX.input1.ID ID – ID włącznika u::Lux	0-1	Aktualne stan wejścia binarnego nr 1.
<b>Zdarzenie</b>	E.ULUX.input1.ID	0-1	Zdarzenie informujące o zmianie stanu wejścia binarnego nr 1.



#### 1. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- **Kanał:** E.ULUX.input.1

**Dane:** 1

Wejście binarne nr 1 podłączone do włącznika o identyfikatorze nr 1 zostało zwarte.

- **Kanał:** E.ULUX.input.1

**Dane:** 0

Wejście binarne nr 1 podłączone do włącznika o identyfikatorze nr 1 zostało rozwarte.

## ULUX.input2

Identyfikator informujący o aktualnym stanie wejścia binarnego nr 2 oraz zmianach jego stanu.

	Identyfikator	Wartość	Opis
<b>Stan</b>	ULUX.input2.ID ID – ID włącznika u::Lux	0-1	Aktualne stan wejścia binarnego nr 2.
<b>Zdarzenie</b>	E.ULUX.input2.ID	0-1	Zdarzenie informujące o zmianie stanu wejścia binarnego nr 2.



### 1. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- **Kanał:** E.ULUX.input.2  
**Dane:** 1  
Wejście binarne nr 2 podłączone do włącznika o identyfikatorze nr 1 zostało zwarte.
- **Kanał:** E.ULUX.input.2  
**Dane:** 0  
Wejście binarne nr 2 podłączone do włącznika o identyfikatorze nr 1 zostało rozwarte.

## 12.10. IFTTT

Identyfikatory IFTTT umożliwiają wysyłanie komend uruchamiających aplety utworzone w serwisie IFTTT oraz odbieranie zdarzeń dla żądań przychodzących do modułu **Base** z serwisu IFTTT.

	Identyfikator	Wartość	Opis
<b>Zdarzenie</b>	<code>E.IFTTT.request</code>	dowone dane	Zdarzenie informujące o żądaniu przychodzącym z serwera IFTTT.
<b>Komenda</b>	<code>C.IFTTT</code>	nazwa_apletu lub nazwa_apletu?dodatkowe_dane	Komenda wywołująca aplet na serwerze IFTTT. Dodatkowe dane mają nast. składnię <code>atrybut1=wartość1;atrybut2=wartość2;...atrybutN=wartośćN</code>

### Komenda IFTTT

Służy do wywoływania apletu o nazwie wskazanej w treści komendy. Składnia komendy umożliwia również przekazanie dodatkowych parametrów do wywoływanej apletu.



Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Wywołanie apletu bez dodatkowych parametrów**

Na potrzeby przykładu założmy, że aplet ma nazwę *test*

**Nazwa:** `C.IFTTT`

**Wartość:** `test`

Wywołanie apletu wskazanego w polu **Wartość**.

- **Wywołanie apletu z dodatkowymi parametrami**

Na potrzeby przykładu założmy, że aplet ma nazwę *temperatura*

**Nazwa:** `C.IFTTT`

**Wartość:** `temperatura;sensor1=24.5;sensor2=20`

Ta komenda wywoła aplet w serwisie IFTTT o nazwie *temperatura* i dodatkowo przekaże parametry: `sensor1` o wartości 24.5 oraz `sensor2` o wartości 20.

### Zdarzenie IFTTT

Zdarzenie jest rozgłaszane w momencie odebrania żądania wysłanego z serwera IFTTT. Zdarzenie może zostać użyte jako inicjator dowolnych akcji w systemie DOMIQ.



- Na potrzeby przykładu założmy, że aplet ma nazwę *test*

**Nazwa:** `E.IFTTT.request`

**Wartość:** `tester`

Odebrano żądanie z serwera IFTTT z wartością `tester`.



## 12.11. Grafana

Identyfikatory tego rodzaju służą do obsługi zdarzeń generowanych przez oprogramowania Grafana w wypadku wystąpienia tzw. alarmów. Ma to miejsce, gdy obserwowane dane spełnią założone kryteria np. przekroczenie określonej temperatury, przekroczenie poziomu cieczy itd. Ogólna składnia zdarzenia jest następująca:

`E.GRAFANA.alert=<nazwa_alarmu>`



- Na potrzeby przykładu założmy, że alarm ma nazwę `temperatura`. Wówczas zdarzenie odebrane przez **Base** może mieć nast. formę:

**Nazwa:** `E.GRAFANA.alert`

**Wartość:** `temperatura`

Odebrano zdarzenie informujące o wystąpieniu alarmu w oprogramowaniu Grafana o nazwie *temperatura*.

## 12.12. UAV

Identyfikatory z grupy UAV przeznaczone są do obsługi urządzeń wykorzystujących protokół UPNP.

Przedstawione przykłady dotyczą integracji DOMIQ z odtwarzaczami SONOS.

W użytych przykładach przyjęliśmy *biuro* jako nazwę urządzenia. Komendy sterujące mogą być wysyłane również jednocześnie do wszystkich urządzeń, wówczas w miejsce nazwy należy wpisać *all*.

	Identyfikator	Wartość	Opis
<b>Stan</b>	UAV.album.N N - nazwa urządzenia	tekst	Nazwa aktualnie odtwarzanego albumu
	UAV.creator.N	tekst	Nazwa wykonawcy aktualnie odtwarzanego utworu.
	UAV.duration.N	tekst	Czas trwania utworu.
	UAV.mode.N	tekst	Tryb odtwarzania (np. normalny, potwarzanie, losowy itd.).
	UAV.state.N	tekst	Aktualny stan odtwarzania: stop/pauza/odtwarzanie.
	UAV.title.N	tekst	Tytuł odtwarzanego utworu.
	UAV.trackno.N	tekst	Numer odtwarzanego pliku.
	UAV.tracks.N	tekst	Całkowita liczba plików w danym folderze.
	UAV.uri.N	tekst	Ścieżka do pliku/strumienia audio (np. radio internetowe).
	UAV.volume.N	tekst	Aktualna głośność.
<b>Zdarzenie</b>	E.UAV.P.N P - nazwa parametr N - nazwa urządzenia	tekst	Zdarzenia są rozgłaszane dla wszystkich zmian stanu parametrów opisanych powyżej.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	C.UAV.volume.N N - nazwa urządzenia	0-100  change:xx	Ustawienie głośności na określoną wartość (0-100%)  Relatywna zmiana głośności względem bieżącej wartości. xx przyjmuje wartości dodatnie lub ujemne (patrz przykłady poniżej).
	C.UAV.mute.N	0/off/1/on	Sterowanie wyciszeniem dźwięku. 1 lub on oznacza wyciszenie.
	C.UAV.control.N	play/stop/pause/next/prev	Funkcja sterująca odtwarzaniem. Dostępne wartości: <ul style="list-style-type: none"> <li>• stop - Zatrzymanie odtwarzania. Odtwarzanie po komendzie stop jest rozpoczynane od początku listy.</li> <li>• play - Rozpoczęcie odtwarzania.</li> <li>• pause - Wstrzymanie odtwarzania.</li> <li>• next - Następny utwór.</li> <li>• prev - Poprzedni utwór.</li> </ul>
	C.UAV.uri.N	<ścieżka>	Wskazanie odtwarzaczowi ścieżki do pliku, który ma być odtworzony. Ścieżką musi wskazywać plik na dysku twardym lub dysku sieciowym. Może być to także adres strumienia radia internetowego lub pliku w serwisie streamingowym. W przypadku plików na dyskach, można wskazać konkretny utwór lub plik z playlistą w formacie m3u. <b>WAŻNE:</b> Wywołanie funkcji uri nie powoduje automatycznego rozpoczęcia odtwarzania. Aby rozpocząć odtwarzanie, należy po funkcji uri, wywołać funkcję control z wartością play.
	C.UAV.group.add.N	<nazwa_urządzenia_nadrzędnego>	Wskazane urządzenie zostanie połączone w grupę ze wskazanym urządzeniem nadrzędnym.
	C.UAV.group.leave.N	<nazwa_urządzenia_nadrzędnego>	Odłączenie wskazanego urządzenia od grupy tworzonej ze wskazanym urządzeniem nadrzędnym.
	C.UAV.notify.N	src:<ścieżka>	Pozwala odtworzyć wybrany plik dźwiękowy bez zakłócenia aktualnie odtwarzanej muzyki lub stacji radiowej. Umożliwia użycie urządzenia SONOS do odtwarzania notyfikacji głosowych lub dźwiękowych w reakcji na dowolne zdarzenia w systemie DOMIQ, np. notyfikacja głosowa o otwartych oknach, gdy wychodzisz z domu i uzbijasz alarm. Możesz też użyć SONOS jako dzwonnka do drzwi.



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Sterowanie głośnością**

**Nazwa:** C.UAV.volume.biuro

**Wartość:** 30

Ustaw głośność urządzenia o nazwie *biuro* na 30%.

**Nazwa:** C.UAV.volume.all

**Wartość:** 30

Ustaw głośność wszystkich urządzeń na 30%.

- **Sterowanie odtwarzaniem**

**Nazwa:** C.UAV.control.biuro

**Wartość:** play

Rozpocznij odtwarzanie w urządzeniu o nazwie *biuro*.

- **Załadowanie wskazanej ścieżki**

**Nazwa:** C.UAV.uri.biuro

**Wartość:** dowolna\_ścieżka

- **Dołączenie do grupy**

**Nazwa:** C.UAV.group.add.biuro

**Wartość:** salon

Połączenie odtwarzacza o nazwie *biuro* w grupę z odtwarzaczem *salon*.

- **Odtworzenie notyfikacji dźwiękowej**

**Nazwa:** C.UAV.notify.all

**Wartość:** src:x-file-cifs://Example/sounds/Ding-dong-chime.mp3

Otworzenie wskazanego dźwięku na wszystkich odtwarzaczach.

### 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- **Kanał:** E.UAV.volume.biuro

**Dane:** 30

Głośność urządzenia *biuro* przyjęła wartość 30%.

- **Kanał:** E.UAV.mute.biuro

**Dane:** 1

Dźwięk w urządzeniu *biuro* został wyciszony.

## 12.13. Połączenia szeregowe

Identyfikatory przedstawione w tym podrozdziale służą do integracji systemu DOMIQ z urządzeniami przy użyciu standardowych interfejsów szeregowych RS-232 lub RS-485.

Do integracji modułu DOMIQ/Base z urządzeniami RS-232 wymagany jest dodatkowo moduł DOMIQ/Serial-2SG. Podobny moduł o oznaczeniu DOMIQ/Serial-4SG służy do łączenia z wykorzystaniem RS-485.

Typowym zastosowaniem łącz RS-232 jest komunikacja z pojedynczym urządzeniem, np. systemem multiroom audio, rzutnikiem multimedialnym itd.

Łąca RS-485 są wykorzystywane bądź ze względu na dużo większą odporność na zakłócenia, bądź z powodu możliwości podłączenia wielu urządzeń równolegle do jednego interfejsu szeregowego.

Poniższa tabela zawiera przegląd wszystkich identyfikatorów przeznaczonych do sterowania modułów szeregowych DOMIQ i ich główne cechy.

Identyfikator	Skrócony opis	Stan	Zdarzenia	Komenda
LC.SER.config	Konfiguracja formatu i prędkości łącza	-	-	+
LC.SER.line	Wysyłanie lub odbiór pełnych linii tekstu	-	+	+
LC.SER.send	Wysyłanie danych	-	-	+

### LC.SER.line

Identyfikator LC.SER.line służy do wysłania dowolnego tekstu do interfejsu RS-232 lub informuje o odebraniu polecenia.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	LC.SER.line.N N - Numer modułu (1,2)	Tekst	Wysłanie dowolnego tekstu do interfejsu szeregowego.
<b>Zdarzenie</b>	LE.SER.line.N	Tekst	Informacja o odebraniu pełnej linii danych.

## 12. Identyfikatory



Przedstawione poniżej przykłady zostały zaczerpnięte z samouczka opisującego proces integracji **DOMIQ** z systemem multiroom audio firmy **NuVo**.

### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Nazwa:** LC.SER.line.1

**Wartość:** \*T'A'FM98.8

Wysłanie komendy do modułu z adresem 1 tekstu o treści \*T'A'FM98.8, co w przypadku integracji z multiroom Nuvo jest przykładem komendy ustawiającej tuner radiowy A na częstotliwość 98.8MHz.

- **Nazwa:** LC.SER.line.1

**Wartość:** \*Z1ON\r\*Z1SRC5\r\*Z1VOL50

Wysłanie komendy do modułu z adresem 1 tekstu o treści \*Z1ON\r\*Z1SRC5\r\*Z1VOL50, co w przypadku integracji z Nuvo jest komendą o nast. treści: włącz stęę pierwszą, przypisz do niej źródło sygnału nr 5, ustaw głośność na 50.

### 2. Zdarzenia na zmianę stanu

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- **Kanał:** LE.SER.line.1

**Dane:** #Z1OFF

Odebrano dane o wartości #Z1OFF, co w przypadku integracji z Nuvo jest informacją o wyłączeniu strefy muzycznej nr 1.

## LC.SER.config

Komenda `LC.SER.config` przeznaczona jest do konfiguracji parametrów komunikacji modułów uniwersalnych interfejsów szeregowych RS-232 lub RS-485. Komenda ma zastosowanie, gdy jest potrzebna dynamiczna zmiana parametrów transmisji. Domyślnie parametry transmisji są ustalane w konfiguratorze w zakładce **Ustawienia**.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<code>LC.SER.config.N</code> N - Numer modułu (1,2)	<b>P F</b> P - Prędkość transmisji: (9600, 19200, 38400, 57600) F - Format ramki danych: (8N1, 8N2, 8O1, 8E1) Uwaga! Pomiędzy prędkością i formatem ramki jest spacja.	Konfiguracja parametrów interfejsu



### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Nazwa:** `LC.SER.config.1`  
**Wartość:** `57600 8N1`  
Ustaw prędkość transmisji na 57600 bodów oraz format ramki danych na 8N1 w module o adresie 1.

## LC.SER.send

Identyfikator `LC.SER.send` służy do wysłania dowolnego tekstu przez interfejs RS-232. W odróżnieniu od `LC.SER.line`, w komendzie `LC.SER.send` nie jest wysyłany znacznik końca linii. Daje to możliwość wysłania kilku niezależnych partii danych (tekstów), jako jedną linię.

Innym sposobem jego użycia jest wysyłanie poleceń binarnych, np. dla integracji z systemami audio etc.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<code>LC.SER.send.N</code> N - Numer modułu (1,2)	Dane	Wysłanie dowolnego tekstu do interfejsu szeregowego.



- Wysłanych zostało pięć niezależnych komend `LC.SER.send`:
  - `LC.SER.send.1=Przykładowa`
  - `LC.SER.send.1=tresc`
  - `LC.SER.send.1=wyslana`
  - `LC.SER.send.1=przez`
  - `LC.SER.send.1=RS232`
- W rezultacie wyświetlony zostanie jeden komunikat: „Przykładowa treść wysłana przez RS232”.
- Wysłanie binarnego ciągu danych w skrypcie w zakładce **Logika**  
`command("LCN.SER.send.1", string.char(0x20, 0x2A, 0x00))`

## 12.14. Komunikacja sieciowa

Identyfikatory opisane w tym podrozdziale umożliwiają wysyłanie danych z użyciem protokołów TCP, UDP oraz HTTP.

### TCP

Identyfikator **TCP** przeznaczony jest do integracji systemu DOMIQ z innymi urządzeniami wykorzystując protokół TCP jako protokół transportowy. Służy do tego komenda `TCP.send`. Dzięki niej możliwe jest np. sterowanie telewizorami, amplitunerami, projektorami itp.

Dane przesyłane są w formie tekstowej. Jeżeli dana integracja wymaga znaku nowej linii, wówczas należy w danych na zakończenie dodać `"\r\n"` (CRLF).

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<code>C.TCP.send.IP:P</code> <b>IP</b> - Adres IP do którego mają zostać wysłane dane. <b>P</b> - Numer portu	Dane	Wysłanie dowolnych danych przy użyciu protokołu TCP



Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Wywołanie apletu bez dodatkowych parametrów**

Na potrzeby przykładu założmy, że aplet ma nazwę `test`

**Nazwa:** `C.TCP.send.192.168.10.100:1234`

**Wartość:** `hello world`

Wysłanie na na IP `192.168.10.100` i port `1234` tekstu `hello world`.

### UDP

Identyfikator **UDP** działa analogicznie do **TCP** z tą różnicą, że protokołem transportowym jest protokół UDP.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<code>C.UDP.send.IP:P</code> <b>IP</b> - Adres IP <b>P</b> - Numer portu	Dane	Wysłanie dowolnych danych przy użyciu protokołu UDP



Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Wywołanie apletu bez dodatkowych parametrów**

Na potrzeby przykładu założmy, że aplet ma nazwę `test`

**Nazwa:** `C.UDP.send.192.168.10.100:1234`

**Wartość:** `hello world`

Wysłanie na na IP `192.168.10.100` i port `1234` tekstu `hello world`.



## HTTP.get

Komenda `HTTP.get` służy do wysyłania żądań HTTP z użyciem metody GET. Komenda nie przetwarza odpowiedzi zwracanej przez serwer (natychmiast po wysłaniu żądania zamykane jest połączenie). Jeżeli jest wymagane przetwarzanie odpowiedzi, wówczas żądania należy wysłać z użyciem modułu TCP opisanego w rozdziale **Logika**.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<code>C.HTTP.get</code>	URL żądania	Wysłanie dowolnego żądania HTTP metodą GET. URL żądania może mieć zarówno formę samej ścieżki, jak również ścieżki z dodatkowymi parametrami.



Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

Na potrzeby przykładu założymy, że żądanie HTTP ma nast. URL: <http://www.example.com/get/data>

- **Żądanie bez dodatkowych parametrów**

**Nazwa:** `C.HTTP.get`

**Wartość:** <http://www.example.com/get/data>

Wysłanie żądania na adres <http://www.example.com/get/data>

Drugi przykład jest bardziej realny - przedstawia sposób na autoresetowanie modułu Base np. w efekcie wywołania timera.

**Nazwa:** `C.HTTP.get`

**Wartość:** <http://admin:admin@192.168.10.102/call/reboot>

Dane uwierzytelniające i adres IP należy zastąpić danymi dostosowanymi do Twojej instalacji.

- **Żądanie z dodatkowymi parametrami**

**Nazwa:** `C.HTTP.get`

**Wartość:** <http://www.example.com/get/data?param1=value1&param2=value2>

Wysłanie żądania na adres <http://www.example.com/get/data> z użyciem dodatkowych parametrów: `param1=value1&param2=value2`

## 12.15. Notyfikacje

Identyfikatory z grupy **Notyfikacje** zaprojektowane zostały z myślą o interakcji systemu DOMIQ z użytkownikiem. Wywołanie identyfikatorów z grupy **Notyfikacje** najczęściej powiązane jest z wystąpieniem innego zdarzenia w systemie automatyki budynkowej.

Poniższa tabela zawiera przegląd wszystkich identyfikatorów z grupy **Notyfikacje** wraz z ich głównymi cechami. Szczegółowy opis poszczególnych identyfikatorów z przykładami zastosowania znajduje się w dalszej części podrozdziału.

Identyfikator	Skrócony opis	Stan	Zdarzenie	Komenda
REMOTE.message	Wyświetlenie dowolnej wiadomości na urządzeniach z aplikacją DOMIQ/Remote, znajdujących się w zasięgu sieci lokalnej, do której podłączony jest moduł Base.	-	-	+
REMOTE.notify	Wyświetlenie dowolnej wiadomości na urządzeniach z aplikacją DOMIQ/Remote.	-	-	+
MAIL	Umożliwia wysłanie e-maila w reakcji na dowolne zdarzenie w systemie DOMIQ.	-	-	+
DISPLAY.screen	Zdarzenie powodujące zmianę ekranu wyświetlanego na wizualizacji	-	+	-
DISPLAY.layer	Zdarzenie powodujące przełączenie widoczności warstwy na wizualizacji	-	+	-

### REMOTE.message

Identyfikator `REMOTE.message` pozwala wyświetlić dowolną wiadomość na ekranach urządzeń mobilnych ze sporowaną aplikacją DOMIQ/Remote, które znajdują się w zasięgu sieci lokalnej, do której podłączony jest moduł DOMIQ/Base.

Aby wiadomość została wyświetlona aplikacja DOMIQ/Remote musi być uruchomiona i podłączona do modułu Base.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<code>C.REMOTE.message</code>	Tekst	Wyświetlenie dowolnej informacji tekstowej.



Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Żądanie bez dodatkowych parametrów**

**Nazwa:** `C.REMOTE.message`

**Wartość:** Alarm w strefie 1

Wyświetl na ekranie wszystkich urządzeń mobilnych z zarejestrowaną aplikacją DOMIQ/Remote, znajdujących się w sieci lokalnej wiadomość o treści: „Alarm w sterfie 1”.

## REMOTE.notify

Identyfikator `REMOTE.notify` umożliwia wyświetlanie notyfikacji na ekranach urządzeń mobilnych ze sparowaną aplikacją DOMIQ/Remote. Notyfikacje są wyświetlane na ekranie nawet, gdy aplikacja Remote nie jest uruchomiona. Notyfikacje dostarczane są za pośrednictwem serwerów Apple (Remote na iOS) i Google (Remote na Android), co nie gwarantuje ich natychmiastowego doręczenia i nie należy traktować ich jako narzędzia do natychmiastowego powiadomienia użytkowników. Wysyłanie notyfikacji można powiązać z innymi zdarzeniami w systemie automatyki domowej.

Notyfikacje mogą być również wysyłane do pojedynczych urządzeń lub do grupy urządzeń przypisanych do wskazanego profilu użytkowników.

Identyfikatory sparowanych urządzeń oraz nazwy profili znajdziesz w konfiguratorze: **Ustawienia** -> **Autoryzacja dla aplikacji**

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<code>C.REMOTE.notify</code>	Tekst	Wyświetlenie dowolnej informacji tekstowej na wszystkich sparowanych urządzeniach jednocześnie.
	<code>C.REMOTE.notify.&lt;id_urządzenia&gt;</code>	Tekst	Wyświetlenie dowolnej informacji tekstowej na wskazanym sparowanym urządzeniu.
	<code>C.REMOTE.notify.&lt;nazwa_profilu&gt;</code>	Tekst	Wyświetlenie dowolnej informacji tekstowej na sparowanych urządzeniach, które są przypisane do wskazanego profilu użytkownika.



Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Notyfikacja do wszystkich urządzeń**

**Nazwa:** `C.REMOTE.notify`

**Wartość:** Witamy w DOMIQ!

Wyświetl na ekranie wszystkich urządzeń mobilnych ze sparowaną aplikacją DOMIQ/Remote wiadomość o treści: „Witamy w DOMIQ!”.

- **Notyfikacja do pojedynczego urządzenia**

**Nazwa:** `C.REMOTE.notify.aabbccddeeff`

**Wartość:** Witamy w DOMIQ!

Wyświetl na ekranie urządzenia od identyfikatorze `aabbccddeeff` wiadomość o treści: „Witamy w DOMIQ!”.

- **Notyfikacja do profilu użytkowników**

**Nazwa:** `C.REMOTE.notify.rodzice`

**Wartość:** Witamy w DOMIQ!

Wyświetl wiadomość o treści: „Witamy w DOMIQ!” na ekranie wszystkich urządzeń przypisanych do profilu `rodzice`.

## 12. Identyfikatory

### EMAIL

Identyfikator `MAIL` umożliwia wysłanie komendy, której rezultatem jest wysłanie e-maila o wskazanym temacie i treści na podany adres. Do prawidłowego działania tej funkcjonalności wymagane jest, aby moduł Base był podłączony do internetu i aby miał aktywną opcję **Sterowanie przez Internet** (zakładka **Ustawienia**).

Adres nadawczy wysyłanych wiadomości ma nast. format `<losowy_identyfikator>@domiq.pl`, przykładowo: `e0f22a9eef91daaaf56ea57c15f1a9ca@domiq.pl`

	Identyfikator	Wartość	Opis
<b>Komenda</b>	<code>SC.EMAIL.&lt;adresat&gt;</code>	<code>temat;treść</code>	Wysłanie emaila na wskazany adres.



Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Nazwa:** `SC.EMAIL.test@test.com`  
**Wartość:** `testowy temat;testowa wiadomość`  
Spowoduje wysłanie wiadomości na adres `test@test.com` z tematem `testowy temat` i wiadomością o treści: `testowa wiadomość`.

### DISPLAY.screen

Zdarzenie, które wymusza wyświetlenie określonego ekranu wizualizacji na wszystkich podłączonych aplikacjach DO-MIQ/Remote, które aktualnie wyświetlają wizualizację. Jest to szczególnie przydatna funkcja w przypadku użycia tabletów jako paneli sterujących. Przykładem może być wyświetlenie podglądu z kamery w reakcji naciśnięcie dzwonka do drzwi/furtki.

	Identyfikator	Wartość	Opis
<b>Zdarzenie</b>	<code>E.DISPLAY.screen</code>	<code>nazwa_ekranu</code>	Zdarzenie informujące podłączone aplikacje o konieczności przełączenia ekranu wizualizacji.



Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Nazwa:** `E.DISPLAY.screen`  
**Wartość:** `kamery`  
Spowoduje rozgłoszenie zdarzenia, które wymusi we wszystkich podłączonych aplikacjach Remote przełączenie ekranu wizualizacji na ekran `kamery`.

### DISPLAY.layer

Zdarzenie, które steruje widocznością warstwy na ekranie wizualizacji, na wszystkich podłączonych aplikacjach DO-MIQ/Remote, które aktualnie wyświetlają wizualizację. Jest to szczególnie przydatna funkcja w przypadku użycia tabletów jako paneli sterujących.

	Identyfikator	Wartość	Opis
<b>Zdarzenie</b>	<code>E.DISPLAY.layer</code>	<code>nazwa_warstwy:0/1</code>	Zdarzenie informujące podłączone aplikacje o konieczności przełączenia widoczności warstwy. wartość 1 - wyświetlenie warstwy wartość 0 - ukrycie warstwy



Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Nazwa:** E.DISPLAY.layers

**Wartość:** ogrzewanie:1

Spowoduje rozgłoszenie zdarzenia, które wymusi we wszystkich podłączonych aplikacjach Remote aktywowanie na wizualizacji warstwy o nazwie ogrzewanie.

## 12.16. Łączy

Identyfikatory z tej grupy służą do wysyłania komend i odbierania zdarzeń pomiędzy modułami Base pracującymi w jednej instalacji.

### Komendy i zdarzenia sieciowe

Komendy sieciowe umożliwiają wykonanie określonego działania w innym module Base. Komendy są poprzedzone przedrostkiem **NC** (ang. **N**etwork **C**ommand).

Zdarzenia sieciowe natomiast umożliwiają odbieranie informacji z innego modułu Base.

	Identyfikator	Wartość	Opis
<b>Komenda</b>	NC . M . K <b>M</b> - Nazwa modułu <b>Base</b> , do którego adresowane jest polecenie. <b>K</b> - Komenda	dowolna	Dowolna komenda sieciowa
<b>Zdarzenie</b>	NE . M . E <b>M</b> - Nazwa modułu <b>Base</b> , w którym wystąpiło zdarzenie. <b>E</b> - Treść zdarzenia.	dowolna	Dowolne zdarzenie sieciowe



Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Komenda sieciowa**

**Kanał:** NC.recepcja.C.VAR.zalanie.mieszkanie1

**Dane:** 1

Ustaw w module o nazwie *recepcja* zmienną *VAR.zalanie.mieszkanie1* na wartość 1.

- **Zdarzenia sieciowe**

**Kanał:** NE.mieszkanie1.E.IDS.alarm.1

**Dane:** 1

Zdarzenie informujące, że w module o nazwie *mieszkanie1* został wzbudzony alarm w strefie 1.

**Kanał:** NE.(%w+).E.IDS.input.10=1

**Dane:** 1

Zdarzenie informujące o zmianie wartości wejścia alarmowego nr 10 w dowolnym module, z którym ustanowione jest połączenie. Wejściem może być np. czujnik pożarowy, kontaktron itd.

## Zmienne sieciowe NET

Identyfikatory z grupy `NET` służą do obsługi zmiennych sieciowych. Każdy moduł Base przypisany do danej grupy ma pełny dostęp do zmiennych sieciowych przynależnych do tej grupy. Jako pełny dostęp należy rozumieć:

- Odczyt stanu.
- Możliwość reagowania na zmiany stanu (zdarzenia).
- Możliwość zmian stanu (komendy).
- Możliwość tworzenia nowych zmiennych sieciowych. Zmienne sieciowe nie wymagają deklaracji. Aby utworzyć nową zmienną sieciową, wystarczy wysłać komendę do nieistniejącej jeszcze zmiennej sieciowej (patrz przykłady poniżej).

	Identyfikator	Wartość	Opis
<b>Stan</b>	<code>NET.G.N</code> <b>G</b> - Nazwa grupy <b>N</b> - Nazwa zmiennej	dowolna	Stan zmiennej sieciowej
<b>Zdarzenie</b>	<code>E.NET.G.N</code>	dowolna	Zdarzenie informujące o zmianie stanu zmiennej sieciowej
<b>Komenda</b>	<code>C.NET.G.N</code>	dowolna	Komenda zmieniająca stan zmiennej sieciowej.



Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

### 1. Komenda

- **Kanał:** `C.NET.parter.wiatr`  
**Dane:** 10

Przypisz wartość 10 do zmiennej sieciowej o nazwie `wiatr`. Zmienna sieciowa przynależy do grupy `parter`. Jeżeli zmienna o nazwie `wiatr` nie istnieje, wówczas zostanie utworzona i zostanie do niej przypisana wartość 10.

### 2. Zdarzenie

- **Kanał:** `E.NET.parter.wiatr`  
**Dane:** 10

Zmienna sieciowa o nazwie `wiatr`, przynależna do grupy `parter` zmieniła wartość na 10.

### Połączenia master-slave (MNT)

Identyfikatory MNT służą do zarządzania połączeniami master-slave między dwoma modułami Base.

Stan modułu podrzędnego (slave) dostępny jest stanie modułu nadrzędnego (master) z przedrostkiem MNT. <nazwa\_połączenia>, gdzie <nazwa\_połączenia>, to nazwa nadana temu połączeniu w zakładce **Łącza**.

	Identyfikator	Wartość	Opis
<b>Stan</b>	MNT.<nazwa_połączenia>.I I - identyfikator z modułu podrzędnego	dowolna wartość	Stan identyfikatora w module podrzędnym
<b>Zdarzenie</b>	E.MNT.<nazwa_połączenia>.I I - identyfikator z modułu podrzędnego	dowolna wartość	Zdarzenie informujące o zmianie stanu w module podrzędnym.
<b>Komenda</b>	C.MNT.<nazwa_połączenia>.I I - identyfikator z modułu podrzędnego	dowolna wartość	Komenda sterująca modulem podrzędnym.

W opisanych poniżej przykładach przyjęliśmy nazwę połączenia `biuro_produkcja`.



#### 1. Komenda sterująca

Poniższe przykłady przedstawiają definicję komend w przypadku użycia ich jako akcji w zdarzeniach/timerach lub poleceniu przypisanym do elementu **Przycisk** w zakładce **Remote**.

- **Nazwa:** `C.MNT.biuro_produkcja.LCN.output.0.10.1`

**Wartość:** 100

Komenda wysteruje wyjście ściemniane nr 1 na 100% w module LCN z adresem 10, który jest podłączony do modułu podrzędnego.

#### 2. Zdarzenia na zmianę stanu w module podrzędnym

Zdarzenie może posłużyć jako inicjator dowolnej sekwencji akcji w systemie.

- **Kanał:** `E.MNT.biuro_produkcja.LCN.output.0.10.1`

**Dane:** 100

Wyjście ściemniane nr 1 w module z adresem 10 osiągnęło wartość 100%.

#### 3. Sterowanie z aplikacji DOMIQ/Remote

Ten przykład stanowi rozwinięcie pierwszego przykładu.

- **Kontrolka Ściemniacz**

W polu **Kanał** wpisz identyfikator, którym chcesz sterować w module podrzędnym:

`MNT.biuro_produkcja.LCN.output.0.10.1`



## 12.17. Zdarzenia systemowe

Ta grupa identyfikatorów `SYSTEM` służy do rozgłaszania zdarzeń, które dotyczą pracy systemu w module Base.

### Restart systemu

To zdarzenie jest rozgłaszane za każdym razem, gdy moduł Base zakończy uruchamianie systemu i jest gotowy do pracy.

	Identyfikator	Wartość	Opis
<b>Zdarzenie</b>	<code>LE.SYSTEM</code>	<code>restarted</code>	Zdarzenie rozgłaszane jest po pełnym uruchomieniu się systemu operacyjnego modułu Base.



- **Kanał:** `LE.SYSTEM`  
**Dane:** `restarted`  
 Moduł Base zakończył uruchamianie systemu operacyjnego.

# 12.18. Identyfikatory własne (makra)

Oprócz predefiniowanych identyfikatorów, system DOMIQ umożliwia tworzenie identyfikatorów przez użytkowników na potrzeby danej funkcjonalności. O tym typie identyfikatorów należy myśleć jako o makrach, które umożliwiają wykonanie określonej, z góry założonej funkcjonalności. Najczęściej przyjmują one formę komend, które sterują wieloma urządzeniami jednocześnie. Nazwy identyfikatorów własnych nie mogą zawierać spacji i polskich znaków.

Podstawową paradygmatem identyfikatorów własnych jest: **zdefiniuj raz, wywołuj z wielu miejsc**. A kiedy pojawi się potrzeba modyfikacji sekwencji czynności robisz to wyłącznie w jego definicji, wszystkie wywołania makra pozostają niezmienione, co znacznie redukuje czas konfiguracji.

Definicja identyfikatorów własnych ma miejsce w zakładce **Zdarzenia**, natomiast odwołania do nich można robić w dowolnych miejscach systemu np. jako akcja timera, jako akcja na naciśnięcie przycisku w aplikacji Remote lub jako efekt działania zdarzenia lub innego makra.



### 1. Makro sterujące wszystkimi roletami na parterze:

Na potrzeby tego przykładu zakładamy, że sterujemy 4 roletami podłączonymi do modułu Satel INT-IORS o adresie 1, zarządzanego przez moduł DOMIQ/Expander.

- Dodaj zdarzenie:

**Kanał:** *C.rolety.parter*

Użyta tu nazwa jest naszą propozycją, możesz użyć innej.

**Dane:** pozostaw to pole puste, aby makro reagowało na dowolną komendę sterującą roletami.

**Akcje -> +Komenda:**

**Nazwa:** *C.EXP.shutter.1.1*

**Wartość:** *\$D0*

Analogicznie dodaj komendy dla pozostałych rolet. Użycie w polu **Wartość** wzorca \$D0 umożliwi przechwycenia wartości komendy *C.rolety.parter* i przekazanie jej do poszczególnych rolet.

Dzięki takiemu podejściu zdefiniowane zostało makro sterujące 4 roletami (komend może być znacznie więcej). To makro może być teraz użyte jako komenda w akcjach timera, jako efekt naciśnięcia przycisku w aplikacji Remote, jako akcja wywołana przez inne zdarzenie lub makro.

### 2. Makro na opuszczenie budynku.

Efekt działania takiego makra może być zmiana nastaw temperatury, obniżenie wydajności wentylacji, wyłączenie światła itd. Jest tu pełna dowolność. Tak utworzone makro może być podobnie, jak w pierwszym przykładzie wywołane w wyniku wywołania zdarzenia lub naciśnięcia przycisku przez użytkownika.

- Dodaj zdarzenie:

**Kanał:** *C.home*

Użyta tu nazwa jest naszą propozycją, możesz użyć innej.

**Dane:** *leave*

Użyta tu wartość jest również naszą propozycją, możesz użyć innej lub np. wartości liczbowej

**Akcje ->** zdefiniuj własną sekwencję czynności na wyjście z budynku.

Analogicznie możesz zdefiniować makro na powrót do domu.

### 3. Zagnieżdżone makra

Jedno makro w swoich akcjach może wywoływać inne makro. Ten przykład jest rozwinięciem przykładu nr 2. Załóżmy, że na wyjście z budynku ma być:

- zmniejszona wydajność wentylacji
- mają być zgaszone wszystkie światła
- mają zostać obniżone temperatury ogrzewania lub wyłączona klimatyzacja w zależności od pory roku.

Gdy przyjmiemy, że każde z tych zadań może być osobnym makrem, wówczas nasze makro na wyjście z budynku może stanowić wywołanie makr na poszczególne działania. Dzięki rozbiciu na mniejsze makra łatwiej jest je modyfikować i system zyskuje większą elastyczność, bo w przyszłości może okazać się potrzebny scenariusz np. samego globalnego odniżenia temperatury lub akcja na zgaszenie wszystkich światła.