

Integracja systemu Ampio z Node-RED

Numer dokumentu: PO-071 Wersja: 12.0 Data publikacji: 13 maja 2025

Opis funkcji Node-RED: Node-RED to narzędzie do programowania przepływu danych, które umożliwia łatwe tworzenie interakcji między różnymi urządzeniami i serwisami. Możesz go wykorzystać do integracji z smart TV, zamkiem TEDEE, itp.

Ostrzeżenia dotyczące ryzyka: Node-RED jest narzędziem programistycznym, które wymaga odpowiedniej wiedzy technicznej. Użytkownik jest odpowiedzialny za zrozumienie logiki programu stworzonego w Node-RED.

Brak wsparcia technicznego dla logiki aplikacji: Ampio nie udziela wsparcia technicznego ani gwarancji co do logiki stworzonych przez użytkowników przepływów w Node-RED. Jest to narzędzie udostępniane na odpowiedzialność instalatora.

Odpowiedzialność za błędy i straty: Ampio nie ponosi odpowiedzialności za błędy w logice programu, ani za ewentualne straty wynikające z korzystania z Node-RED.

Zachęta do testowania: Zachęcamy do przeprowadzenia testów na niewielką skalę przed pełnym wdrożeniem projektu opartego na Node-RED. To pomoże zminimalizować potencjalne ryzyka.

Dostarczenie pomocnych źródeł wiedzy: Udostępniamy dokumentację Node-RED oraz inne materiały edukacyjne, które mogą pomóc w zrozumieniu i efektywnym korzystaniu z narzędzia.

Prosimy o przeczytanie instrukcji ze zrozumieniem przed rozpoczęciem korzystania z Node-RED. Jeśli masz jakiekolwiek pytania, skonsultuj się z działem wsparcia technicznego.

Uruchamianie interfejsu Node-RED

Lokalnie

Aby skorzystać z możliwości integracji systemu Ampio oraz platformy Node-RED należy znać adres IP serwera oraz hasło użytkownika admin i uruchomić interfejs przez przeglądarkę.

Na module M-SERV znajdziemy interfejs platformy Node-RED na porcie 1880, wpisując w przeglądarkę IP:1880 (np. 192.168.1.2:1880).

Przez chmurę

Aby uruchomić Node-RED przez chmurę, generujemy numer seryjny i hasło z poziomu aplikacji Ampio UNI. Klikamy na logo na dole ekranu, wybieramy opcję *Wsparcie zdalne*, i klikamy START.



Następnie znajdujemy w przeglądarce internetowej dowolną stronę z MD5 Hash Generator i wpisujemy wygenerowany w aplikacji numer seryjny i hasło jednym ciągiem znaków.

MD5 Hash Generator

Use this generator to	o create an MD5 hash of a string:
40895puvhk8	
	4
Generate →	
Your String	40895puvhk8
MD5 Hash	3d50f1fc210584ff16a569009f0ee989 Copy
SHA1 Hash	b453f008d5e46cc4927d1a69c14ea7e105106622 Copy

Kopiujemy wygenerowany MD5 Hash.

Aby dostać się do Node-RED zdalnie musimy znać następujący link:

https://node-[wygenerowany MD5 Hash].ampio.pl

W naszym przypadku:

https://node-3d50f1fc210584ff16a569009f0ee989.ampio.pl

Taki adres wklejamy w naszą przeglądarkę:

••• •	■ node-3d50f1fc21058	4ff16a569009f0ee989.ampio.pl	¢	٩	Ĉ +	- ©
Node-RED						
			_			
			_			
			۴~			
		Password:				
	Node-BED		Login			
	noue nub					

Integracja przez gotowe bloczki

Wybór bloczków Node-RED znajduje się z lewej strony interfejsu, mamy tam między innymi 4 predefiniowane bloczki Ampio, które ułatwią nam tworzenie zależności.



Każdy z bloczków posiada swój własny poradnik w zakładce help.



Aby upewnić się czy biblioteka node-red-contrib-ampio jest aktualna należy wejść w menu 🛛 Manage palette i sprawdzić czy jest możliwy update. Jeżeli aktualizacja zostanie wykonana, należy zresetować serwer.

User Settings			
			Close
View	Nodes	Install	
Keyboard	Q filter nodes		
Palette	 1.2.9 > 46 nodes 		in use
	 node-red-contrib 0.4.6 5 nodes 	-ampio	remove disable all

Przy dodawaniu pierwszego bloczka z biblioteki Ampio warto zrobić *Deploy* i sprawdzić czy bloczek otrzymał status *connected*. Następnie otwieramy menu bloczka, wybieramy ikonę lupy i sprawdzamy czy w ciągu 15 sekund pojawi się lista naszych urządzeń.

	Properties		
Ampio IN connected	Server	Ampio Config	✓
	Target device	0001 - MSERV-3s ()	~ Q
	Value Type	Digital output	~

Jeżeli bloczek nie ma statusu connected może być konieczne skonfigurowanie go. Wybieramy edycję obok pola Server (ikona ołówka). W zakładce Connection w polu Server wpisujemy localhost, port zostawiamy 1883, natomiast w zakładce Credentials podajemy dane do logowania takie jak do interfejsu www serwera Ampio.

Delete				Cancel	Update
Properties					•
Name Name	Name				
Connection	1	Credentials			
Server	localhost		Port	1883	

Następnie wciskamy Update, Done oraz robimy Deploy.

Testowanie działania kodu

W platformie Node-RED istnieje możliwość debugowania kodu i wyświetlania danych w polu z prawej strony interfejsu. Przykład na zrzucie:

- pomarańczowy nazwa Flow
- czerwony włączenie zakładki debug
- niebieski bloczek pozwalający na wyświetlanie informacji w debug
- fioletowy dane wyświetlone w opcji debug

	Successfully deployed				=/	Deplo	y 🔻		2		
Flow 1	Successfully injected: timestamp			+	ji€ debug	i		¥ ₹all	nodes		•
‡ tim	iestamp	msg.payload)		10/19/2021, 9:00 msg.payload : n 16346268099	0:09 AM umber 597	node: c4	4ce8b(6.dd8588)	•

Bloczek Ampio IN -- pobieranie informacji z wejść

Edit Ampio IN no	de			
Delete			Cancel	Done
Properties				\$
Server address	localhost			
Target device	BD57 - MDC	DT-M4 ()	~	Q
Value Type	Digital input			~
I/O ID	1:			~
Name	Name			
Ignore Retain Flag				

Parametry:

- 1. Server address adres serwera, jeżeli Node-RED jest bezpośrednio na serwerze, wybieramy Ampio Config lub localhost
- 2. Target device urządzenie w sieci CAN z unikalnym adresem MAC
- 3. Value Type służy do wyboru typu wartości, jaką otrzymujemy z urządzenia. Wybieramy z listy spośród typów:
 - Analog value-wartości analogowe
 - RGB color- kolor oświetlenia RGB
 - Digital input-wejście cyfrowe
 - Digital output- wyjście cyfrowe
 - Flag flaga
 - Temperature-wartości temperatury
- 4. I/O ID numer wejścia lub wyjścia urządzenia
- 5. Opcjonalnie możemy nadać nazwę blokowi w polu Name.

Pod prawidłowo skonfigurowanym blokiem Ampio IN pojawi się napis connected.

Bloczek Ampio OUT -- sterowanie wyjściami

Delete		Cancel Done
Properties		
Server address	localhost	
Target device	BD57 - MDOT-M4 ()	~ Q
Cmd type	Standard	~
I/O ID	1:	~
Name	Name	

Parametry:

- Server address adres serwera, jeżeli Node-RED jest bezpośrednio na serwerze, wybieramy Ampio Config lub localhost
- 2. Target device urządzenie w sieci CAN z unikalnym adresem MAC
- 3. Cmd Type służy do wyboru typu wartości, jaką mamy zamiar kontrolować. Wybieramy z listy spośród typów:
 - · Standard wyjścia binarne lub analogowe
 - RAW CAN broadcast Typ RAW służy do obsługi surowych danych zawartych w osobnej dokumentacji sieci CAN.
 - Digital input-wejście cyfrowe
 - Digital output- wyjście cyfrowe
 - Flag flaga
 - Temperature-wartości temperatury
- 4. I/O ID numer wejścia lub wyjścia urządzenia
- 5. Opcjonalnie możemy nadać nazwę blokowi w polu Name.

Pod prawidłowo skonfigurowanym blokiem Ampio OUT pojawi się napis connected.

Bloczek Ampio LCD -- wyświetlanie informacji na wyświetlaczach paneli M-DOT

Edit Ampio LCD r	node	
Delete	Cancel Do	ne
Properties	۵	Þ
Server address	localhost	
Target device	BD57 - MDOT-M4 ()	
LCD font size	Standard (10x16)	
LCD X position	0	
LCD Y position	0	
LCD text color		
LCD bg color (hex)	000000	
Name	Name	
O Enabled		

Parametry:

- 1. Server address adres serwera, jeżeli Node-RED jest bezpośrednio na serwerze, wybieramy Ampio Config lub localhost
- 2. Target device urządzenie w sieci CAN z unikalnym adresem MAC
- 3. LCD font size wybór czcionki spośród Standard (10×16), Big (20×32) oraz Icon (40×40). Ostatni typ dostępny jest, jeżeli do panelu wgrane są ikony.
- 4. LCD X position początek tekstu, współrzędna X
- 5. LCD Y position początek tekstu, współrzędna Y
 6. LCD text color kolor tekstu zapisany w systemie szesnastkowym
- 7. LCD bg color kolor tła zapisany w systemie szesnastkowym
- 8. Name opcjonalna nazwa

Punktem początkowym jest lewy górny róg ekranu. Jego współrzędne to LCD X=00 i LCD Y=00. Wraz ze zwiększaniem wartości LCD X, tekst będzie coraz bliżej prawej krawędzi. Zwiększenie wartości LCD Y skutkuje obniżeniem tekstu, schemat poniżej.



Bloczek Ampio Event -- przesyłanie zdarzeń z lub do systemu Ampio

Edit Ampio Event	s node				
Delete			Cancel		Done
Properties				٥	l į
Server address	localhost				
Name	Name				

Parametry:

- 1. Server address adres serwera, jeżeli Node-RED jest bezpośrednio na serwerze, wybieramy localhost
- 2. Name opcjonalna nazwa

Przykład użycia

Pobranie temperatury z modułu Ampio i wyświetlenie go wraz z opisem na wyświetlaczu panelu M-DOT.

W bloczku Ampio IN ustawiamy z jakiego modułu pobieramy temperaturę, typ danych oraz numer czujnika.

Delete		Cancel Done
Properties		
Server	Ampio Config	~
Target device	0011 - MSERV-3s ()	~ Q
Value Type	Temperature	~
I/O ID		~
Name	Name	
Ignore Retain Flag		

W bloczku funkcyjnym dodajemy opis używając funkcji języka Javascript.

Edit function no	de				
Delete				Cancel	Done
Properties					
Name 💊	Name				
Setup		Function	Close		
					×*
1 msg.pa 2 return	yload = "TEMP: msg;	" + msg.payload;			

W bloczku Ampio LCD wybieramy moduł M-DOT ,ustawiamy w którym miejscu ma się pojawić napis, rozmiar i kolor czcionki oraz kolor tła.

Edit Ampio LCD r	node	
Delete		Cancel Done
Properties		
Server	Ampio Config	✓
Target device	9EEF - MDOT-6LCD ()	~ Q
LCD font size	Standard (10x16)	~
LCD X position	5	
LCD Y position	5	
LCD text color (hex)	FFFFFF	
LCD bg color (hex)	000000	
Name	Name	

Bloczki łączymy i klikamy deploy.



W ten sposób otrzymujemy na środku wyświetlacza biały napis "TEMP: wartość temperatury" na czarnym tle.

Integracja przez broker MQTT

Istnieje możliwość integracj platformy Node-RED z systemem Ampio bez użycia gotowych bloczków. W tym celu należy użyć *brokera MQTT*. W ten sposób można śledzić więcej wydarzeń w sieci CAN oraz kontrolować bardziej zaawansowane funkcje.

W tym celu należy użyć przede wszystkim bloczków mątt in oraz mątt out.

 network 				
))	mqtt in			
mqtt out				

Jeżeli używamy Node-RED zainstalowanego na module M-SERV, w ustawieniach serwera wpisujemy *localhost*. W innym wypadku parametry logowania są identyczne jak do aplikacji Smart Home Manager a serwer podajemy za pomocą adresu IP. Po dokonaniu zmian należy nacisnąć Add, Done a następnie Deploy.

Edit mqtt in node > Add new mqtt-broker config node					
				Cancel	Add
Properties					
Name Name	Name				
Connection		Security	Messag	es	
Server	localhost		Port 1883		
Enable secure (SSL/TLS) connection					
Client ID	Leave blank	k for auto generated			
 Keep alive tin 	ne (s) 60	✓ Use clean session			
Use legacy M	IQTT 3.1 supp	port			

MQTT topic

Komunikacja z brokerem MQTT jest przede wszystkim oparta na dwóch kierunkach (*from* i *to*). Topic zaczynający się od ampio/from umożliwia sprawdzenie stanów urządzeń. Topic zaczynający się od ampio/to służy do sterowania urządzeniami.

Przykłady z użyciem ampio/to:

ampio/to/<mac>/cmd - wykonaj komendę

ampio/to/<mac>/raw - surowa ramka danych

ampio/to/can/dev/list - lista urządzeń w sieci CAN

ampio/to/event - wyślij zdarzenie

Lista komend:

Komenda	Торіс	Payload
ustaw pojedyncze wyjście RGB RGBW roleta flagi MRT- temperatura MRT-temeratura dzień/noc MRT- tryb pracy	<pre>ampio/to/<mac>/o/<nr>/cmd ampio/to/<mac>/rgb/<nr>/cmd ampio/to/<mac>/rgbw/<nr>/cmd ampio/to/<mac>/o/<nr>/cmd ampio/to/<mac>/f/<nr>/cmd ampio/to/<mac>/rs/<nr>/cmd ampio/to/<mac>/rsdn/<nr>/cmd ampio/to/<mac>/rgb/<nr>/cmd ampio/to/<mac>/rgb/<nr>/cmd</nr></mac></nr></mac></nr></mac></nr></mac></nr></mac></nr></mac></nr></mac></nr></mac></nr></mac></pre>	on,off; 0255 off; 0255,0255,0255 (on – jak ma działać) off; 0255,0255,0255,0255 (on – jak ma działać) 0 – STOP; 1 – DOWN; 2 – UP on,off; 0255 -99.9155.0 temperatura_dzień, temperatura_noc (przykład: 19,20) 0 – kalendarz; 1 – MANUAL; 2 – MANUAL2; 3 – wakacje;

Topic ampio/from przykłady:

Тур	Торіс	Payload string	Przykładowy payload
temperatury	ampio/from/ <mac>/state/t/<nr></nr></mac>	-99.9 do 1000.0 (nawet więcej)	21.5
binary input	<pre>ampio/from/<mac>/state/i/<nr></nr></mac></pre>	0 lub 1	1

Тур	Торіс	Payload string	Przykładowy payload	
binary output	<pre>ampio/from/<mac>/state/o/<nr></nr></mac></pre>	0 lub 1	1	
analog input	ampio/from/ <mac>/state/a/<nr></nr></mac>	0 do 255	0	
binary input extended	ampio/from/ <mac>/state/bi/<nr></nr></mac>	0 lub 1	0	
binary output extended	ampio/from/ <mac>/state/bo/<nr></nr></mac>	0 lub 1	0	
RGBW	ampio/from/ <mac>/state/rgbw/<nr></nr></mac>	0255,0255,0255,0 228 ,220,13,50		
RGB	<pre>ampio/from/<mac>/state/rgb/<nr></nr></mac></pre>	0255,0255,0255	128,220,13	
flagi	<pre>ampio/from/<mac>/state/f/<nr></nr></mac></pre>			
flagi liniowe 8 bit	<pre>ampio/from/<mac>/state/afu8/<nr></nr></mac></pre>	0255		
flagi liniowe 16 bit	<pre>ampio/from/<mac>/state/afi16/<nr></nr></mac></pre>	-3276832767		
MRT – temperatura	<pre>ampio/from/<mac>/state/rs/<nr></nr></mac></pre>	25.5		
wartości analogowe 8bit (DALI, I FD)	<pre>ampio/from/<mac>/state/au/<nr></nr></mac></pre>	0255	234	
wartości analogowe 16bit (ze znakiem)	ampio/from/ <mac>/state/au16/<nr></nr></mac>	065536		
wartości analogowe 16 bit pomniejszone przez 10K	ampio/from/ <mac>/state/au16l/<nr></nr></mac>	06553.6	23.4	
wartości 32 bit np. MODBUS	ampio/from/ <mac>/state/au32/<nr></nr></mac>	04 294 967 296	1234	

Jeżeli nie jest napisane jakim typem danych należy przesłać wartości, informacje przesyłane są typem string.

MQTT test

Jeżeli chcemy sprawdzić wszystkie dane jakie są dostępne w naszym systemie, należy użyć bloczka *mqtt in* i ustawić topic ampio/# podłączając debug. Następnie naciskamy Done oraz Deploy. Dane powinny być widoczne w oknie debug.

Edit mqtt in node				
Delete	Cancel Done			
Properties	* E			
Server	localhost:1883 🗸			
📰 Торіс	ampio/#			
⊛ QoS	2 ~			
G Output	auto-detect (string or buffer)			
♥ Name	Name			

Jeżeli użytkownik chce sprawdzić wszystkie dane z jednego modułu, wystarczy ustawić topic na ampio/from/MAC/# (np. ampio/from/ABCD/#).

MAC adres zawsze podajemy bez zer wiodących.

Sprawdzanie stanów

Do sterowania modułów, zawsze używamy małego (lokalnego) MAC adresu.

Jeżeli użytkownik chce sprawdzić stan pola w module M-DOT, ustawiamy topic jako: ampio/from/3910/state/i/1 gdzie:

- from oznacza kierunek,
- 3910 to adres MAC,
- state oznacza sprawdzenie aktualnego stanu,
- 1 oznacza numer wejścia (M-DOT pole 1)

Sterowanie wyjść

Aby włączyć pierwsze wyjście w urządzenie o MAC adresie ABCD, ustawiamy topic jako: ampio/to/ABCD/o/1/cmd a payload jako: on.

Obie opcje na poniższym rysunku dają ten sam efekt:



Sterowanie M-DOT

Poza sterowaniem wyjściami poprzez komendy, istnieje możliwość tworzenia własnych tekstów na wyświetlaczu M-DOT przy użyciu komend API. Aby z nich skorzystać należy wybrać topic RAW. Konieczne może być zaktualizowanie oprogramowania w module M-DOT.

Sterowanie poprzez API dzielimy w związku z różnymi wielkościami wyświetlaczy LCD:

Termin API użyty jest w tym poradniku w kontekście wysyłania komend z brokera MQTT do wyświetlacza panelu dotykowego. Nie jest to interfejs API związany z komendami HTTP.

M-DOT-M18 oraz M-DOT-M6 (od pcb 8) -- rozdzielczość 240×320

Komendy do wyświetlania (dla wyświetlaczy wieloekranowych):

- 1E 01 zmień numer ekranu (eg. 1E 01 05 zmień ekran na nr 6)
- 1E 02 xx yy yy...- ustaw znaki (mała linia), ekran nr xx na napis yy yy...., max 12 znaków
- 1E 03 xx yy yy...- ustaw znaki (duża linia), ekran nr xx na napis yy yy...., max 9 znaków

Powyższe funkcje wypisujące znaki, działają tylko na ekranach typu *Cztery ikony i treść*. Aby działały prawidłowo, pola *Aktor* w ustawieniach danego ekranu nie mogą być puste.

Oraz funkcje niezależne od ekranów, nadpisujące zawsze aktualny ekran:

• 29 03 – wyczyść ekran

- 29 14 xa xa ya ya xs xs ys ys ff ff cc cc rysuj ramkę na ekranie, (xa xa początek w osi x, ya ya początek w osi y, xs xs rozmiar x, ys ys rozmiar y, ff ff szerokość ramki, cc cc kolor ramki)
- 29 0C xa xa xz xz ya ya yz yz cc cc rysuj prostokąt na ekranie, (xa xa początek w osi x, xz xz koniec w osi x, ya ya początek w osi y, yz yz koniec w osi y, cc cc kolor ramki)
- 29 18 xx xx yy yy cc cc bb bb zz zz zz zz zz rysuj znaki czcionką 40×64 (xx xx początek w osi x, yy yy początek w osi y, cc cc kolor znaków, bb bb kolor tła, zz max. 6 znaków, tylko znaki 0-9 oraz \$, . / : + -)
- 29 13 xx xx yy yy cc cc bb bb zz zz zz zz zz rysuj znaki czcionką 26×48 (xx xx początek w osi x, yy yy początek w osi y, cc cc kolor znaków, bb bb kolor tła, zz max. 9 znaków)
- 29 12 xx xx yy yy cc cc bb bb zz zz zz zz zz rysuj znaki czcionką 20×32 (xx xx początek w osi x, yy yy początek w osi y, cc cc kolor znaków, bb bb kolor tła, zz max. 12 znaków)
- 29 11 xx xx yy yy cc cc bb bb zz zz zz zz zz rysuj znaki czcionką 10×16 (xx xx początek w osi x, yy yy początek w osi y, cc cc kolor znaków, bb bb kolor tła, zz max. 24 znaki)
- 29 15 xa xa ya ya cc cc bb bb ic rysuj ikonę (z pamięci M-DOT, wgrywane przez konfigurator), ic numer ikony, xa start w osi x, ya start w osi y, cc cc kolor ikony, bb bb kolor tła

Na przykład wysłanie: msg.topic = ampio/to/ABAB/raw msg.payload = "1E03003031303130313031";

zmienia duży wiersz na 1 i 0 (zgodnie z kodem ASCII) na M-DOT o MAC adresie ABAB.

Wysłanie msg.payload = "291800000003344AABB30313233" wypisze 0123 na ekranie w określonych kolorach.

Uwaga na kolejność bajtów – starszy/młodszy. Kolor kodowany jest w standardzie RGB565, przy czym przy kodowaniu kolorów z pomocą zewnętrznych generatorów, może być konieczna zmiana kolejności bajtów.

Wysłanie 291251000001... zacznie pisać od pozycji 81 w x i 256 w y. Do przeliczania warto używać kalkulatora Windows w trybie programisty (hex 51 daje dziesiętnie 81, hex 100 daje dziesiętnie 256).

Wysłanie 290C0000EF006A00D30000000 spowoduje namalowanie czarnego prostokąta na całą szerokość ekranu między 1/3 a 2/3 wysokości (jest to drugi wiersz dla opcji 3 wiersze treści).

Edit inject node			
Delete		Са	ncel Done
Properties			
Name	Name		
≡ msg. payl	oad =		
≡ msg. topic	=	✓ ^a _z ampio/to/BD2A/raw	×

Pozostałe M-DOT z mniejszym wyświetlaczem -- rozdzielczość 128×160

- 29 01 07 xx yy cc cc bb bb nn zz zz zz.. ustaw znaki mała linia, max 11 znaków (xx-pozycja x, yy-pozycja y, cc cc kolor znaków, bb bb kolor tła, nn ilość znaków, zz zz ... znaki w formacie Windows-1250)
- 29 01 09 xx yy cc cc bb bb nn zz zz zz.. ustaw znaki duża linia, max 6 znaków (xx-pozycja x, yy-pozycja y, cc cc – kolor znaków, bb bb – kolor tła, nn – ilość znaków, zz zz ... – znaki w formacie Windows-1250)
- 29 01 03 czyszczenie ekranu
- 29 01 04 xa ya xz yz c1 c2 rysowanie prostokąta o zadanym kolorze (xa początek w osi x, ya początek w osi y, xz koniec w osi x, yz koniec w osi y, c1 + c2 kolor prostokąta 2-bajtowo)
- 29 01 0A ic xa ya cc cc bb bb rysuj ikonę (z pamięci M-DOT, wgrywane przez konfigurator), ic numer ikony, xa start w osi x, ya – start w osi y, cc cc – kolor ikony, bb bb – kolor tła

Kolory 2 bajtowe są zapisywane w formacie RGB565.

Multisensor M-SENS

Kolejność danych w multisensorze M-SENS:

- Wilgotność: ampio/from/<mac>/state/au161/1
- Ciśnienie bezwzględne: ampio/from/<mac>/state/au161/2
- Głośność ampio/from/<mac>/state/au161/3
- Jasność ampio/from/<mac>/state/au161/4
- Jakość powietrza: ampio/from/<mac>/state/au161/5
- Ciśnienie względne: ampio/from/<mac>/state/au161/6
- Temperatura: ampio/from/<mac>/state/t/1

Sterowanie raw

Część funkcji sterujących z poziomu Node-RED udostępiona jest za pomocą funkcji RAW. Topic do sterowania przedstawia się następująco: ampio/to/mac_adress/raw.

Poniżej lista przykładowych funkcji, którymi można sterować za pomocą topicu RAW:

Sterowanie flagami binarnymi czasowe

Opis payload:

- D0 0x01 sterowanie flagami
- D1 0x00 sterowanie flagami czasowe
- D2 FLAG_MASK[0] maska flag
- D3 FLAG_MASK[1] maska flag
- D4 FLAG_MASK[2] maska flag
- · D5 FLAG_MASK[3] maska flag
- D6 wartość do ustawienia 0-wyłącz, 1..255 włącz
- D7 TIME[0] czas włączenia/wyłączenia [10ms] dla czasu 0 pozostaje włączona/wyłączona na stałe
- D8 TIME[1] czas włączenia/wyłączenia [10ms] dla czasu 0 pozostaje włączona/wyłączona na stałe
- D9 TIME[2] czas włączenia/wyłączenia [10ms] dla czasu 0 pozostaje włączona/wyłączona na stałe

Przykład: włącza flagę numer 1 w urządzeniu o MAC=30EE, na 2 sek

topic: ampio/to/30EE/raw payload: 010001000000FFC80000

Sterowanie RGBW i RGB proste

Opis payload:

- · D0 0x02 sterowanie wyjściami
- · D1 0x00 sterowanie wyjściami proste RGBW
- D2 R
- D3 G
- D4 B
- D5 W

Przykład: ustaw zadany kolor ustawić: mac=33CD, kolor R=255, G=255, B=0, W=0;

topic: ampio/to/33CD/raw payload: 0200FFFF0000

LUB (np. dla M-SERV-s)

Opis payload:

- D0 0x33 sterowanie wyjściami
- D1 0xDE
- D2 R
- D3 G
- D4 B
- D5 W

Przykład: ustaw zadany kolor ustawić: mac=1, kolor R=255, G=255, B=0, W=0;

topic: ampio/to/1/raw payload: 33DEFFFF0000

Sterowanie flagami liniowymi 8-bitowymi bez znaku (0-255)

Opis payload:

- 7A flagi liniowe 8bit
- F9 ustaw numer na wartość
- 80 wartość na którą ma zostać ustawiona w systemie szesnastkowym
- 00 numer flagi 0..17

Przykład: Ustaw flagę 0 na wartość 128.

topic: ampio/to/1907/raw payload: 7AF98000 (string)

Sterowanie flagami liniowymi 16-bitowymi bez znaku (0-65535)

Opis payload:

- 79 flagi liniowe 16bit
- F2 ustaw numer na wartość
- 34 wartość bajt młodszy na którą ma zostać ustawiona
- 12 wartość bajt starszy na którą ma zostać ustawiona
- 00 numer flagi 0

Przykład: Ustaw flagę 0 na wartość 4660.

topic: ampio/to/1907/raw payload: 79F2341200 (string)

Sterowanie roletami w modułach z możliwością asocjacji (przekaźniki/rolety)

Opis payload:

- · 31 sterowanie roletą
- F9 ustaw numer na wartość
- 02 otwórz (00 stop, 01 zamknij, 02 otwórz)
- 00 numer rolety od 0

Przykład: Otwórz roletę pierwszą.

topic: ampio/to/1907/raw payload: 31F90200 (string)

Nadpisywanie stanu rolety bez jej wysterowania

Opis payload:

- · 31 sterowanie roletą
- E0 roleta (E1 lamelka)
- 02 numer rolety (numerowanie od 0)
- 00 procent rolety (0-100%)

Przykład: Roleta numer 3 na 0 procent.

topic: ampio/to/1907/raw payload: 31E00200 (string)

Wysłanie w magistralę temperatury z zewnętrznego czujnika (z zadanym MAC adresem z zakresu 0-fff)

Funkcja działa od wersji brokera MQTT 4.25.1.

Opis payload:

• 22 - przykład temperatury

Przykład: Wyślij w magistralę temperaturę 22 stopnie jako adres 11001055.

topic: ampio/to/broadcast/55/t payload: 22 (number)

Należy pamiętać że nie wszystkie moduły wspierają wspomniane funkcjonalności. Zalecamy najpierw sprawdzić w konfiguratorze Ampio Designer, czy moduł wspiera np. obsługę flag 16-bitowych.