

Moduł GATE MODBUS

Uwaga!

Opisana funkcjonalność oraz integracja jest dostępna dla **GRENTON GATE MODBUS MASTER, DIN, Eth (INT-201 - E-01)** posiadający **firmware 1.4.1 - 2334 lub wyższy!**

1. Informacje ogólne

Moduł GATE MODBUS umożliwia integrację systemu Grenton ze wszystkimi urządzeniami wspierającymi standard MODBUS RTU / TCP.

Uwaga!

Dla tworzonych obiektów wirtualnych nie ma ograniczenia względem ilości obiektów - ograniczeniem jest pamięć urządzenia, na którą wpływa m.in. poziom rozbudowania logiki na module.

2. Konfiguracja modułu

Uwaga!

Przed rozpoczęciem jakiegokolwiek pracy z modułem GATE Modbus wymagana jest aktualizacja bazy interfejsów!

2.1. Ustawianie czasu za pomocą serwera NTP

Moduł GATE Modbus umożliwia ustawianie czasu za pomocą serwera NTP wraz z uwzględnieniem strefy czasowej a także zmianą czasu (zimowy/letni). Czas pobierany jest automatycznie z serwera NTP (pool.ntp.org).

Do konfiguracji służą trzy cechy:

- `UseNTP` - określa czy GATE używa NTP,
- `NTPTimeout` - czas oczekiwania na odpowiedź z serwera NTP,
- `TimeZone` - ustawianie strefy czasowej GATE - dostępne są 22 strefy.

Uwaga!

Pobieranie czasu z serwera NTP wymaga, aby GATE znajdowało się w sieci, która posiada połączenie z internetem.

Uwaga!

W momencie ustawienia cechy `UseCloud = true`, cecha `UseNTP` jest automatycznie ustawiana na wartość `true`.

3. Obiekty wirtualne

3.1. Protokół Modbus RTU

Uwaga!

GATE MODBUS może pracować w tym samym czasie tylko jako Master **lub** Slave. Po dodaniu obiektu `ModbusSlaveConfigRTU` i wysłaniu konfiguracji GATE MODBUS pracuje wyłącznie jako Slave obsługując obiekty `ModbusSlaveRTU`. Utworzone obiekty wirtualne `ModbusRTU` są ignorowane.

A. ModbusRTU

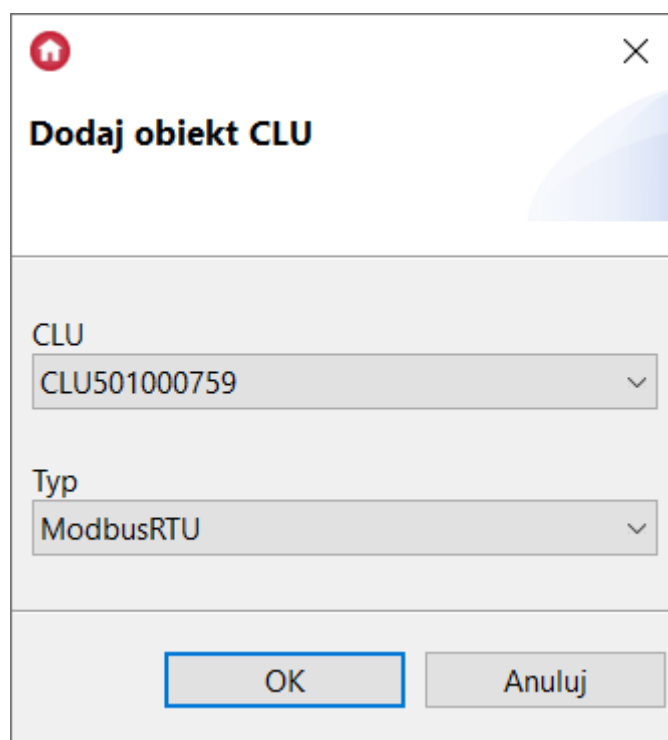
Obiekt wirtualny `ModbusRTU` służy do odczytywania wartości z urządzeń Slave za pomocą protokołu RTU.

Uwaga!

Obiekt wirtualny `ModbusRTU` zastępuje wycofywane obiekty `Modbus` oraz `ModbusValue`.

Aby odczytać wartość z dostępnego urządzenia Slave należy kolejno:

- Utworzyć obiekt wirtualny `ModbusRTU` i nadać mu nazwę:



The image shows a dialog box titled "Dodaj obiekt CLU". It contains two dropdown menus. The first is labeled "CLU" and has the value "CLU501000759" selected. The second is labeled "Typ" and has the value "ModbusRTU" selected. At the bottom of the dialog, there are two buttons: "OK" and "Anuluj".

- Przejsć do zakładki *Cechy wbudowane* i wprowadzić:
 - **TransmissionSpeed** - prędkość transmisji;
 - **Parity** - parametr ustawia sprawdzanie parzystości:
 - 0 - None
 - 1 - Odd
 - 2 - Even
 - **StopBits** - liczba bitów stopu urządzenia Slave:
 - 0 - 1 bit stopu
 - 1 - 1.5 bitu stopu
 - 2 - 2 bity stopu
 - **DeviceAddress** - adres urządzenia Slave;
 - **RefreshInterval** - okres odpytywania rejestru urządzenia Slave przez GATE Modbus;
 - **ResponseTimeout** - czas urządzenia Slave na odpowiedź (po jego przekroczeniu zwracany jest `ErrorCode = -2`);

- **RegisterAddress** - adres obsługiwanej rejestr;
- parametry odpowiednie dla wybranego typu rejestr urządzenia Slave;

Właściwości obiektu
✕

Nazwa:
Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
TransmissionSpeed	19200	<input type="text" value="19200"/>	bps	1200,2400,4800,9600,19200,38400,57600
Parity	0	<input type="text" value="None"/>		0,1,2
StopBits	0	<input type="text" value="1"/>	bits	0,1,2
DeviceAddress	111	<input type="text" value="111"/>		[0-255]
ResponseTimeout	300	<input type="text" value="300"/>	ms	[25-6400]
RefreshInterval	1000	<input type="text" value="1000"/>	ms	[0-300000]
RegisterAddress	23	<input type="text" value="23"/>		[0-65535]
RegisterType	2	<input type="text" value="Rejestry pamiętające (holding)"/>		0,1,2,3
AlwaysWriteMultiple	0	<input type="text" value="No"/>		0,1
InputOutputCount	0	<input type="text" value="1"/>		[1-64]
Data Type	0	<input type="text" value="Unsigned Integer"/>		0,1,2
DataWidth	16	<input type="text" value="16"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian (SwapBytesAndWor)"/>		0,1,2,3
BitFieldWidth	16	<input type="text" value="0"/>	bits	[0-64]
BitFieldPosition	0	<input type="text" value="0"/>	bit	[0-63]
Divisor	1	<input type="text" value="1"/>		
InitialValueAccess	0	<input type="text" value="Read"/>		0,1
InitialValue	-	<input type="text" value="0"/>		
Value	100			
RawValue	25600			
IsValueValid	1		bool	
ErrorCode	0			

Auto odświeżanie
 Odśwież

Uwaga!

W obiekcie Gate Modbus od wersji 1.4.1 - 2334 została wprowadzona cecha `ModbusMasterFrameSpace` służąca do określania przerwy między wysyłanymi ramkami Modbus wyrażonej w znakach. W przypadku projektów z dużą ilością obiektów `ModbusRTU` i problemów z płynnością odpytywania dla krótkiego czasu `RefreshInterval` można wprowadzić od 1 do 50 pustych znaków między wysyłanymi ramkami.

B. ModbusSlaveConfigRTU

Obiekt wirtualny `ModbusSlaveConfigRTU` służy do konfiguracji modułu Gate Modbus pracującego jako urządzenie Slave. Aby tego dokonać należy:

- Utworzyć obiektu wirtualny `ModbusSlaveConfigRTU` :

Dodaj obiekt CLU

CLU
 CLU501000759

Typ
 ModbusSlaveConfigRTU

OK Anuluj

- Przejść do zakładki *Cechy wbudowane* i wprowadzić dane dotyczące komunikacji z urządzeniem Master RTU:
 - **TransmissionSpeed** - prędkość transmisji;
 - **Parity** - parametr ustawia sprawdzanie parzystości:
 - 0 - None
 - 1 - Odd
 - 2 - Even
 - **StopBits** - liczba bitów stopu urządzenia Slave:
 - 0 - 1 bit stopu
 - 1 - 1.5 bitu stopu
 - 2 - 2 bity stopu

Po wysłaniu konfiguracji Gate Modbus pracuje jako urządzenie Slave.

Właściwości obiektu

Nazwa: ModbusConfig Typ: ModbusSlaveConfigRTU
 Id: CLU501000746->MOD1737

Sterowanie Zdarzenia Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
TransmissionSpeed	19200	19200	bps	1200,2400,4800,9600,19200,38400,57600
Parity	0	None		0,1,2
StopBits	0	1	bits	0,1,2

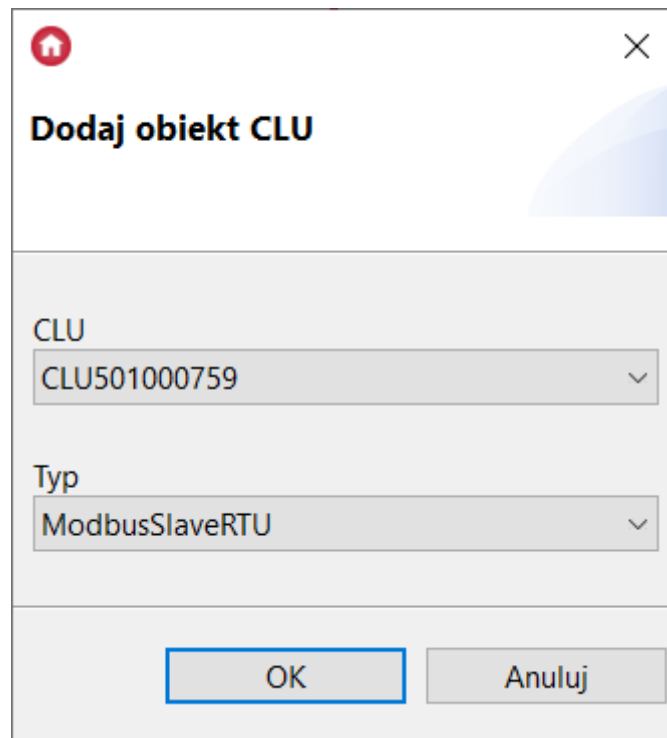
Auto odświeżanie

OK Anuluj

C. ModbusSlaveRTU

Obiekt wirtualny `ModbusSlaveRTU` służy do definiowania wartości dla poszczególnych rejestrów urządzenia Slave. Aby obiekt działał prawidłowo w pierwszej kolejności należy utworzyć obiekt `ModbusSlaveConfigRTU` ([patrz pkt.B.](#)). W celu skorzystania z funkcjonalności należy:

- Utworzyć obiekt wirtualny `ModbusSlaveRTU` :



The image shows a software dialog box titled "Dodaj obiekt CLU". It features a title bar with a home icon on the left and a close button (X) on the right. The main area contains two dropdown menus. The first is labeled "CLU" and has the value "CLU501000759" selected. The second is labeled "Typ" and has the value "ModbusSlaveRTU" selected. At the bottom of the dialog, there are two buttons: "OK" and "Anuluj". The "OK" button is highlighted with a blue border.

- Przejść do zakładki *Cechy wbudowane* i wprowadzić dane dotyczące komunikacji z urządzeniem Master RTU:
 - **DeviceAddress** - adres urządzenia Slave;
 - **RegisterAddress** - adres obsługiwanego rejestru;
 - **RegisterType** - typ rejestru;
 - **DataType** - typ wartości;
 - **DataWidth** - szerokość danych;
 - **Endianness** - kolejność bajtów;
 - **InitialValue** - początkowa wartość rejestru.

Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
DeviceAddress	111	<input type="text" value="111"/>		[0-255]
RegisterAddress	23	<input type="text" value="23"/>		[0-65535]
Register Type	2	Rejestry pamiętające (holding)		0,1,2,3
Data Type	0	Unsigned Integer		0,1,2
Data Width	16	<input type="text" value="16"/>	bits	16,32,48,64
Endianness	0	Big Endian (SwapBytesAndWords)		0,1,2,3
Initial Value	-	<input type="text" value="100"/>		
Value	100			
Raw Value	25600			

Auto odświeżanie

D. Modbus

Uwaga!

Obiekt wirtualny `Modbus` został wycofany w wersji Gate Modbus 1.4.1 - 2334 wraz z wprowadzeniem nowego obiektu wirtualnego `ModbusRTU`, który jest jego odpowiednikiem. W celu zapewnienia kompatybilności z już istniejącymi projektami obiekt pozostaje dostępny do użytku. Możliwość tworzenia nowych obiektów tego typu została wyłączona.

Chcąc przeprowadzić poprawną konfigurację modułu Gate Modbus należy kolejno:

- Utworzyć obiekt wirtualny `Modbus` i nadać mu nazwę:

Wybierz obiekt

Wybierz CLU:

Obiekt:

- Przejść do zakładki `Cechy wbudowane` i wprowadzić:
 - **DeviceAddress** - adres urządzenia Slave;

- **AccessRights** - tryb pracy (*Read* - odczyt wartości z rejestru; *ReadWrite* - umożliwia zapis wartości do ustawionego rejestru);
- **RegisterAddress** - adres obsługiwanego rejestru;
- **TransmissionSpeed** - prędkość transmisji;
- **RefreshInterval** - okres odpytywania rejestru urządzenia Slave przez GATE Modbus;
- **ResponseTimeout** - czas urządzenia Slave na odpowiedź (po jego przekroczeniu zwracany jest `ErrorCode = -2`);
- **Divisor** - dzielnik (dla `ValueType = Number/Float`);
- parametry odpowiednie dla wybranego typu rejestru urządzenia Slave;
- **StopBits** - liczba bitów stopu urządzenia Slave:
 - 0 - 1 bit stopu
 - 1 - 1.5 bitu stopu
 - 2 - 2 bity stopu
- **Parity** - parametr ustawia sprawdzanie parzystości:
 - 0 - None
 - 1 - Odd
 - 2 - Even

×

Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
DeviceAddress	111	<input type="text" value="111"/>	number	[0-255]
AccessRights	0	<input type="text" value="Read"/>	-	0,1
RegisterAddress	141	<input type="text" value="141"/>	number	[0-65535]
TransmissionSpeed	9600	<input type="text" value="9600"/>	bps	1200,2400,4800,9600,19200,38400,57600,115200
ValueType	1	<input type="text" value="Number"/>		1,2,3
BitPosition	0	<input type="text" value="0"/>	number	[0-15]
BitCount	16	<input type="text" value="16"/>	number	[1-32]
RefreshInterval	1000	<input type="text" value="1000"/>	number	[0-65535]
ResponseTimeout	100	<input type="text" value="100"/>	number	[10-65535]
Divisor	1	<input type="text" value="1"/>	number	[1-65535]
Endianess	3	<input type="text" value="SwapWords"/>	-	0,1,2,3
RegisterType	2	<input type="text" value="HoldingRegisters"/>	-	0,1,2,3
ErrorCode	0		number	
Value	639	<input type="text" value="0"/>	number	
RegisterValue	639		number	
StopBits	0	<input type="text" value="1"/>	-	0,1,2
Parity	0	<input type="text" value="None"/>	-	0,1,2

Auto odświeżanie

- Wysłać konfigurację oraz zweryfikować połączenie - zakładka *Cechy wbudowane*, cecha `ErrorCode = 0` (poprawny odczyt/zapis):


Właściwości obiektu


Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
DeviceAddress	111	<input type="text" value="111"/>	number	[0-255]
AccessRights	0	<input type="text" value="Read"/>	-	0,1
RegisterAddress	141	<input type="text" value="141"/>	number	[0-65535]
TransmissionSpeed	9600	<input type="text" value="9600"/>	bps	1200,2400,4800,9600,19200,38400,57600,115200
ValueType	1	<input type="text" value="Number"/>	-	1,2,3
BitPosition	0	<input type="text" value="0"/>	number	[0-15]
BitCount	16	<input type="text" value="16"/>	number	[1-32]
RefreshInterval	1000	<input type="text" value="1000"/>	number	[0-65535]
ResponseTimeout	100	<input type="text" value="100"/>	number	[10-65535]
Divisor	1	<input type="text" value="1"/>	number	[1-65535]
Endianness	3	<input type="text" value="SwapWords"/>	-	0,1,2,3
RegisterType	2	<input type="text" value="HoldingRegisters"/>	-	0,1,2,3
ErrorCode	0	<input type="text" value="0"/>	number	
Value	639	<input type="text" value="0"/>	number	
RegisterValue	639		number	
StopBits	0	<input type="text" value="1"/>	-	0,1,2
Parity	0	<input type="text" value="None"/>	-	0,1,2

Auto odświeżanie 

 Odśwież

Uwaga!

W obiekcie Gate Modbus od wersji 1.4.1 - 2334 została wprowadzona cecha `ModbusMasterFrameSpace` służąca do określania przerwy między wysłanymi ramkami Modbus wyrażonej w znakach. W przypadku projektów z dużą ilością obiektów i problemów z płynnością odpytywania dla krótkiego czasu `RefreshInterval` można wprowadzić od 1 do 50 pustych znaków między wysłanymi ramkami.

E. ModbusValue

Uwaga!

Obiekt wirtualny `ModbusValue` został wycofany w wersji Gate Modbus 1.4.1 - 2334 wraz z wprowadzeniem nowego obiektu wirtualnego `ModbusRTU`, który jest jego odpowiednikiem. W celu zapewnienia kompatybilności z już istniejącymi projektami obiekt pozostaje do użytku. Nie jest możliwe tworzenie nowych obiektów tego typu.

Aby użyć obiektu wirtualnego `ModbusValue` należy:

- Utworzyć obiekt wirtualny `ModbusValue` i nadać mu nazwę:

- Przejść do zakładki *Cechy wbudowane* i wprowadzić:
 - **TransmissionSpeed** - prędkość transmisji;
 - **Parity** - parametr ustawia sprawdzanie parzystości:
 - 0 - None
 - 1 - Odd
 - 2 - Even
 - **StopBits** - liczba bitów stopu urządzenia Slave:
 - 0 - 1 bit stopu
 - 1 - 1.5 bitu stopu
 - 2 - 2 bity stopu
 - **DeviceAddress** - adres urządzenia Slave;
 - **ResponseTimeout** - czas na odpowiedź w krokach 25ms;
 - **RefreshPeriod** - minimalny okres odświeżania w krokach 5ms - wartość 0 wyłącza automatyczne odświeżanie;
 - **RegisterAddress** - adres obsługiwanego rejestru;
 - **RegisterType** - typ ustawionego rejestru:
 - 0 - wyjścia dwustanowe
 - 1 - wejścia dwustanowe
 - 2 - rejestry pamiętające
 - 3 - rejestry wejściowe
 - **Divisor** - dzielnik (skala) wartości;
 - **InitialValueAccess** - metoda pierwszego dostępu do wartości Value:
 - 0 - wartość początkowa Value jest odczytywana z urządzenia;
 - 1 - wartość początkowa Value jest zapisywana do urządzenia;

Uwaga!

Wpisanie wybranej wartości w polu `Value` przy ustawieniu cechy `InitialValueAccess` = 1 powoduje zapisanie wartości (wysłanie odpowiedniej ramki) po wysłaniu konfiguracji na CLU. Jeśli wartość cechy `InitialValueAccess` = 0 wpisana wartość w polu `Value` jest ignorowana.

- parametry odpowiednie dla wybranego typu rejestru urządzenia Slave;



Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
TransmissionSpeed	9600	<input type="text" value="9600"/>	bps	1200,2400,4800,9600,19200,38400,57600,115200
Parity	0	<input type="text" value="None"/>		0,1,2
StopBits	0	<input type="text" value="1"/>	bits	0,1,2
DeviceAddress	111	<input type="text" value="111"/>		[0-255]
ResponseTimeout	1000	<input type="text" value="1000"/>	ms	[25-6400]
RefreshPeriod	1000	<input type="text" value="1000"/>	ms	[0-300000]
RegisterAddress	122	<input type="text" value="122"/>		[0-65535]
RegisterType	2	<input type="text" value="Holding registers"/>		0,1,2,3
InputOutputCount	0	<input type="text" value="1"/>		[1-64]
Data Type	0	<input type="text" value="Unsigned Integer"/>		0,1,2
DataWidth	16	<input type="text" value="16"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian"/>		0,1,2,3
BitFieldWidth	16	<input type="text" value="0"/>	bits	[0-64]
BitFieldPosition	0	<input type="text" value="0"/>	bit	[0-63]
Divisor	1	<input type="text" value="1"/>		
InitialValueAccess	0	<input type="text" value="Read"/>		0,1
Value	192	<input type="text" value="0"/>		
RawValue	49152			
IsValid	true		bool	
ErrorCode	0			

Auto odświeżanie 
 Odśwież

- Wysłać konfigurację oraz zweryfikować połączenie - zakładka *Cechy wbudowane*, cecha `ErrorCode` =0 (poprawny odczyt/zapis):

Właściwości obiektu
✕

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
TransmissionSpeed	9600	<input type="text" value="9600"/>	bps	1200,2400,4800,9600,19200,38400,57600,115200
Parity	0	<input type="text" value="None"/>		0,1,2
StopBits	0	<input type="text" value="1"/>	bits	0,1,2
DeviceAddress	111	<input type="text" value="111"/>		[0-255]
ResponseTimeout	1000	<input type="text" value="1000"/>	ms	[25-6400]
RefreshPeriod	1000	<input type="text" value="1000"/>	ms	[0-300000]
RegisterAddress	122	<input type="text" value="122"/>		[0-65535]
RegisterType	2	<input type="text" value="Holding registers"/>		0,1,2,3
InputOutputCount	0	<input type="text" value="1"/>		[1-64]
DataType	0	<input type="text" value="Unsigned Integer"/>		0,1,2
DataWidth	16	<input type="text" value="16"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian"/>		0,1,2,3
BitFieldWidth	16	<input type="text" value="0"/>	bits	[0-64]
BitFieldPosition	0	<input type="text" value="0"/>	bit	[0-63]
Divisor	1	<input type="text" value="1"/>		
InitialValueAccess	0	<input type="text" value="Read"/>		0,1
Value	192	<input type="text" value="0"/>		
RawValue	49152			
IsValid	true		bool	
ErrorCode	0			

Auto odświeżanie

Uwaga!

W obiekcie Gate Modbus od wersji 1.4.1 - 2334 została wprowadzona cecha `ModbusMasterFrameSpace` służąca do określania przerwy między wysłanymi ramkami Modbus wyrażonej w znakach. W przypadku projektów z dużą ilością obiektów i problemów z płynnością odpytywania dla krótkiego czasu `RefreshInterval` można wprowadzić od 1 do 50 pustych znaków między wysłanymi ramkami.

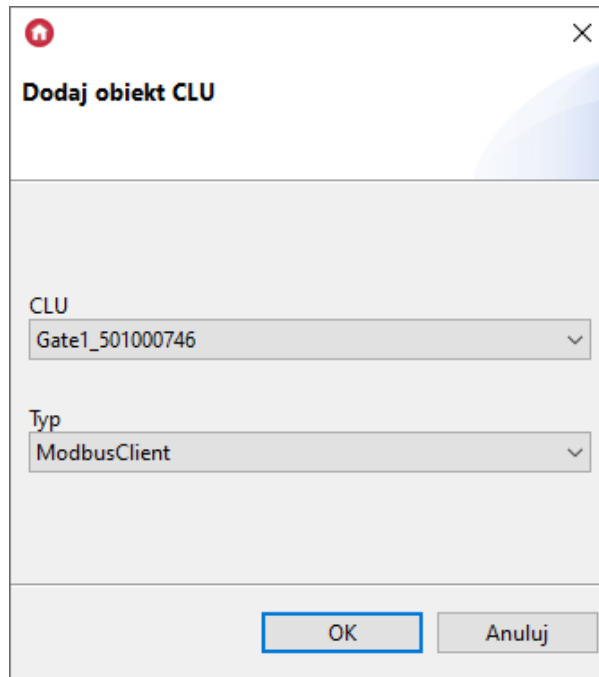
3.2. Protokół Modbus TCP

A. ModbusClient

Obiekt wirtualny `ModbusClient` służy do komunikacji z urządzeniami typu serwer za pomocą sieci LAN.

Aby użyć obiektu wirtualnego `ModbusClient` należy:

- Utworzyć obiekt wirtualny `ModbusClient` i nadać mu nazwę:



The screenshot shows a dialog box titled "Dodaj obiekt CLU". It contains two dropdown menus. The first is labeled "CLU" and has the value "Gate1_501000746". The second is labeled "Typ" and has the value "ModbusClient". At the bottom of the dialog, there are two buttons: "OK" and "Anuluj".

- Przejść do zakładki *Cechy wbudowane* i wprowadzić:
 - **SocketAddress** - adres IP urządzenia Modbus;
 - **DeviceAddress** - adres urządzenia Modbus;
 - **ResponseTimeout** - czas na odpowiedź w krokach 25ms;
 - **RefreshInterval** - minimalny okres odświeżania w krokach 5ms - wartość 0 wyłącza automatyczne odświeżanie;
 - **RegisterAddress** - adres obsługiwanego rejestru;
 - **RegisterType** - typ ustawionego rejestru:
 - 0 - wyjścia dwustanowe
 - 1 - wejścia dwustanowe
 - 2 - rejestry pamiętające
 - 3 - rejestry wejściowe
 - **Divisor** - dzielnik (skala) wartości;
 - **InitialValueAccess** - metoda pierwszego dostępu do wartości Value:
 - 0 - wartość początkowa Value jest odczytywana z urządzenia
 - 1 - wartość początkowa Value jest zapisywana do urządzenia
 - parametry odpowiednie dla wybranego typu rejestru urządzenia serwer:

Uwaga!

Wpisanie wybranej wartości w polu `InitialValue` przy ustawieniu cechy `InitialValueAccess` = 1 powoduje zapisanie wartości (wysłanie odpowiedniej ramki) po wysłaniu konfiguracji na CLU. Jeśli wartość cechy `InitialValueAccess` = 0 wpisana wartość w polu `InitialValue` jest ignorowana.

- Wysłać konfigurację oraz zweryfikować połączenie - zakładka *Cechy wbudowane*, cecha `ErrorCode` = 0 (poprawny odczyt/zapis):

Właściwości obiektu
✕

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
SocketAddress	192.168.0.3	<input type="text" value="192.168.0.3"/>		
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
ResponseTimeout	1500	<input type="text" value="1500"/>	ms	[25-6400]
RefreshInterval	1000	<input type="text" value="1000"/>	ms	[0-300000]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	<input type="text" value="Rejestry pamiętające (ho"/>		0,1,2,3
AlwaysWriteMultiple	0	<input type="text" value="Yes"/>		0,1
InputOutputCount	0	<input type="text" value="1"/>		[1-64]
DataType	0	<input type="text" value="Unsigned Integer"/>		0,1,2
DataWidth	16	<input type="text" value="16"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian (SwapBytesAr"/>		0,1,2,3
BitFieldWidth	16	<input type="text" value="0"/>	bits	[0-64]
BitFieldPosition	0	<input type="text" value="0"/>	bit	[0-63]
Divisor	1	<input type="text" value="1"/>		
InitialValueAccess	0	<input type="text" value="Read"/>		0,1
InitialValue	-	<input type="text" value="59900"/>		
Value	1103			
RawValue	20228			
IsValueValid	1		bool	
ErrorCode	0			

Auto odświeżanie

Uwaga!

Domyślny port pracy urządzenia serwer wynosi 502. Istnieje możliwość pracy na innym, zdefiniowanym porcie. W takim wypadku po wpisaniu adresu IP urządzenia należy po dodać docelowy port komunikacji - przykładowo: .

B. ModbusServer

Obiekt wirtualny służy do komunikacji z urządzeniami typu klient za pomocą sieci LAN.

Aby użyć obiektu wirtualnego należy:

- Utworzyć obiekt wirtualny i nadać mu nazwę:

Dodaj obiekt CLU

CLU
CLU12345692

Typ
ModbusServer

OK Anuluj

- Przejść do zakładki *Cechy wbudowane* i wprowadzić:
 - **Port** - Port nasłuchiwania serwera;
 - **DeviceAddress** - adres urządzenia Modbus;
 - **RegisterAddress** - adres obsługiwanego rejestru;
 - **RegisterType** - typ ustawionego rejestru:
 - 0 - wyjścia dwustanowe
 - 1 - wejścia dwustanowe
 - 2 - rejestry pamiętające
 - 3 - rejestry wejściowe
 - **DataType** - typ wartości;
 - **DataWidth** - szerokość danych;
 - **Endianness** - kolejność bajtów;
 - **InitialValue** - początkowa wartość rejestru;
- Wysłać konfigurację oraz zweryfikować połączenie za pomocą urządzenia klient

Właściwości obiektu
✕

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
Port	502	<input type="text" value="502"/>		
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	Rejestry pamiętające (ho		0,1,2,3
DataType	0	Unsigned Integer		0,1,2
DataWidth	16	<input type="text" value="16"/>	bits	16,32,48,64
Endianness	0	Big Endian (SwapBytesAr		0,1,2,3
InitialValue	-	<input type="text" value="1996"/>		
Value	1996			
RawValue	52231			

Auto odświeżanie

4. Parametry rejestrów

Uwaga!

Opisane obiekty wirtualne dostępne są dla **GRENTON GATE MODBUS MASTER, DIN, Eth (INT-201 - E-01)** posiadający **firmware 1.4.1 - 2334 lub wyższy!**

4.1. Obiekt ModbusRTU oraz ModbusClient

Parametry rejestrów w przypadku obiektów `ModbusRTU` oraz `ModbusClient` nie różnią się w obszarze konfiguracji - jedyna różnica polega na sposobie komunikacji z docelowym urządzeniem. Poniżej przedstawiono sposoby odczytu wartości z rejestrów urządzeń Slave / serwer.

A. 16-bitowe wartości całkowite rejestrów

Odczyt 16-bitowych rejestrów pamiętających (`Read Holding Registers`, `FunctionCode = 03`):

- `RegisterType`: Holding Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 16;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: wartość domyślna;
- `BitFieldPosition`: wartość domyślna.

Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
TransmissionSpeed	19200	<input type="text" value="19200"/>	bps	1200,2400,4800,9600,19200,38400
Parity	0	<input type="text" value="None"/>		0,1,2
StopBits	0	<input type="text" value="1"/>	bits	0,1,2
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
ResponseTimeout	300	<input type="text" value="300"/>	ms	[25-6400]
RefreshInterval	1000	<input type="text" value="1000"/>	ms	[0-300000]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	<input type="text" value="Rejestry pamiętające (hc)"/>		0,1,2,3
AlwaysWriteMultiple	0	<input type="text" value="No"/>		0,1
InputOutputCount	0	<input type="text" value="1"/>		[1-64]
DataType	0	<input type="text" value="Unsigned Integer"/>		0,1,2
DataWidth	16	<input type="text" value="16"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian (SwapBytesA)"/>		0,1,2,3
BitFieldWidth	16	<input type="text" value="0"/>	bits	[0-64]
BitFieldPosition	0	<input type="text" value="0"/>	bit	[0-63]
Divisor	1	<input type="text" value="1"/>		
InitialValueAccess	0	<input type="text" value="Read"/>		0,1
InitialValue	-	<input type="text" value="0"/>		
Value	1996			
RawValue	52231			

Auto odświeżanie

Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
SocketAddress	192.168.0.3	<input type="text" value="192.168.0.3"/>		
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
ResponseTimeout	1500	<input type="text" value="1500"/>	ms	[25-6400]
RefreshInterval	1000	<input type="text" value="1000"/>	ms	[0-300000]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	<input type="text" value="Rejestry pamiętające (ho"/>		0,1,2,3
AlwaysWriteMultiple	0	<input type="text" value="No"/>		0,1
InputOutputCount	0	<input type="text" value="1"/>		[1-64]
Data Type	0	<input type="text" value="Unsigned Integer"/>		0,1,2
DataWidth	16	<input type="text" value="16"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian (SwapBytesAr"/>		0,1,2,3
BitFieldWidth	16	<input type="text" value="0"/>	bits	[0-64]
BitFieldPosition	0	<input type="text" value="0"/>	bit	[0-63]
Divisor	1	<input type="text" value="1"/>		
InitialValueAccess	0	<input type="text" value="Read"/>		0,1
InitialValue	-	<input type="text" value="0"/>		
Value	1996			
RawValue	52231			
IsValueValid	1		bool	
ErrorCode	0			

Auto odświeżanie

Odczyt 16-bitowych rejestrów wejściowych (`Read Input Registers`, `FunctionCode = 04`):

- `RegisterType` : Input Registers;
- `DataType` : Unsigned Integer;
- `DataWidth` : 16;
- `Endianness` : wartość domyślna;
- `BitFieldWidth` : wartość domyślna;
- `BitFieldPosition` : wartość domyślna.

Zapis 16-bitowych rejestrów pamiętających (`Preset/Write Single Holding Register`, `FunctionCode = 06`):

- `RegisterType` : Holding Registers;
- `AlwaysWriteMultiple` : No;
- `DataType` : Unsigned Integer;
- `DataWidth` : 16;
- `Endianness` : wartość domyślna;
- `BitFieldWidth` : wartość domyślna;
- `BitFieldPosition` : wartość domyślna.

Uwaga!

W przypadku zapisu do rejestrów 16-bitowych możliwe jest użycie funkcji 16 (`Write Multiple Holding Register`) zamiast funkcji 6 (`Write Single Holding Register`) w przypadku kiedy urządzenie Slave akceptuje jedynie zapis tego typu. W tym celu należy użyć cechy `AlwaysWriteMultiple` i ustawić ją na wartość 1.

Zapis 16-bitowych rejestrów pamiętających (`Preset/Write Multiple Holding Registers`, `FunctionCode = 16`):

- `RegisterType` : Holding Registers;
- `AlwaysWriteMultiple` : Yes
- `DataType` : Unsigned Integer;
- `DataWidth` : 16;
- `Endianness` : wartość domyślna;
- `BitFieldWidth` : wartość domyślna;
- `BitFieldPosition` : wartość domyślna.

Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie Zdarzenia Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
SocketAddress	192.168.0.3	<input type="text" value="192.168.0.3"/>		
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
ResponseTimeout	1500	<input type="text" value="1500"/>	ms	[25-6400]
RefreshInterval	1000	<input type="text" value="1000"/>	ms	[0-300000]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	Rejestry pamiętające (ho		0,1,2,3
AlwaysWriteMultiple	1	Yes		0,1
InputOutputCount	0	<input type="text" value="1"/>		[1-64]
DataType	0	Unsigned Integer		0,1,2
DataWidth	16	<input type="text" value="16"/>	bits	16,32,48,64
Endianness	0	Big Endian (SwapBytesAr		0,1,2,3
BitFieldWidth	16	<input type="text" value="0"/>	bits	[0-64]
BitFieldPosition	0	<input type="text" value="0"/>	bit	[0-63]
Divisor	1	<input type="text" value="1"/>		
InitialValueAccess	0	Read		0,1
InitialValue	-	<input type="text" value="0"/>		
Value	1996			
RawValue	52231			
IsValueValid	1		bool	
ErrorCode	0			

Auto odświeżanie

B. Pola bitowe w rejestrach 16-bitowych

Odczyt pól bitowych w 16-bitowym rejestrze pamiętającym (`Read Holding Registers`, `FunctionCode = 03`):

- `RegisterType` : Holding Registers;
- `DataType` : Unsigned Integer;
- `DataWidth` : 16;
- `Endianness` : wartość domyślna;
- `BitFieldWidth` : 0 - 16 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = `DataWidth`);
- `BitFieldPosition` : 0 - 15 (pozycja pierwszego interesującego bitu).

Uwaga!

Zakres liczbowy cechy `BitFieldWidth` zależy od ustawienia wartości cechy `DataWidth`.

Dla `DataWidth` = 16 zakres cechy `BitFieldWidth` wynosi [0 - 16].

W chwili ustawienia cechy `BitFieldWidth` na wartość równą 0 i wysłaniu konfiguracji cecha przyjmuje wartość maksymalną dla aktualnie ustawionej wartości cechy `DataWidth`.

Uwaga!

Zakres liczbowy cechy `BitFieldPosition` zależy od ustawienia wartości cechy `DataWidth`.

Dla `DataWidth` = 16 zakres cechy `BitFieldPosition` wynosi [0 - 15].

Uwaga!

Cechy `BitFieldWidth` i `BitFieldPosition` są uzależnione od cechy `DataWidth`.

Działają one wg zależności: `BitFieldWidth` + `BitFieldPosition` <= `DataWidth`

Na przykład:

W chwili ustawienia `DataWidth` i `BitFieldWidth` = 16 oraz `BitFieldPosition` = 15 cecha `BitFieldWidth` automatycznie przyjmie wartość = 1.

Dla `BitFieldWidth` = 0 cecha `BitFieldPosition` przyjmuje zawsze 0.

Właściwości obiektu
✕

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
TransmissionSpeed	19200	<input type="text" value="19200"/>	bps	1200,2400,4800,9600,19200,38400
Parity	0	<input type="text" value="None"/>		0,1,2
StopBits	0	<input type="text" value="1"/>	bits	0,1,2
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
ResponseTimeout	300	<input type="text" value="300"/>	ms	[25-6400]
RefreshInterval	1000	<input type="text" value="1000"/>	ms	[0-300000]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	<input type="text" value="Rejestry pamiętające (ho"/>		0,1,2,3
AlwaysWriteMultiple	0	<input type="text" value="No"/>		0,1
InputOutputCount	0	<input type="text" value="1"/>		[1-64]
DataType	0	<input type="text" value="Unsigned Integer"/>		0,1,2
DataWidth	16	<input type="text" value="16"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian (SwapBytesAr"/>		0,1,2,3
BitFieldWidth	1	<input type="text" value="1"/>	bits	[0-64]
BitFieldPosition	2	<input type="text" value="2"/>	bit	[0-63]
Divisor	1	<input type="text" value="1"/>		
InitialValueAccess	0	<input type="text" value="Read"/>		0,1
InitialValue	-	<input type="text" value="0"/>		
Value	1			
RawValue	52231			
IsValueValid	1		bool	
ErrorCode	0			

Auto odświeżanie

Odczyt pól bitowych w 16-bitowym rejestrze wejściowym (`Read Input Registers`, `FunctionCode = 04`):

- `RegisterType` : Input Registers;
- `DataType` : Unsigned Integer;
- `DataWidth` : 16;
- `Endianness` : wartość domyślna;
- `BitFieldWidth` : 0 - 16 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = `DataWidth`);
- `BitFieldPosition` : 0 - 15 (pozycja pierwszego interesującego bitu).

Zapis pól bitowych w 16-bitowym rejestrze pamiętającym (`Preset/Write Single Holding Register`, `FunctionCode = 06`):

- `RegisterType` : Holding Registers;
- `DataType` : Unsigned Integer;
- `DataWidth` : 16;

- `Endianness` : wartość domyślna;
- `BitFieldWidth` : 0 - 16 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = `DataWidth`);
- `BitFieldPosition` : 0 - 15 (pozycja pierwszego interesującego bitu).

C. 32 - bitowe wartości całkowite rejestrów

Odczyt 32 - bitowych wartości całkowitych rejestru pamiętającego (`Read Holding Registers`, `FunctionCode = 03`):

- `RegisterType` : Holding Registers;
- `DataType` : Unsigned Integer;
- `DataWidth` : 32;
- `Endianness` : Big Endian;
- `BitFieldWidth` : wartość domyślna;
- `BitFieldPosition` : wartość domyślna.

Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
TransmissionSpeed	19200	<input type="text" value="19200"/>	bps	1200,2400,4800,9600,19200,3840
Parity	0	<input type="text" value="None"/>		0,1,2
StopBits	0	<input type="text" value="1"/>	bits	0,1,2
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
ResponseTimeout	300	<input type="text" value="300"/>	ms	[25-6400]
RefreshInterval	1000	<input type="text" value="1000"/>	ms	[0-300000]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	<input type="text" value="Rejestry pamiętające (ho"/>		0,1,2,3
AlwaysWriteMultiple	0	<input type="text" value="No"/>		0,1
InputOutputCount	0	<input type="text" value="1"/>		[1-64]
DataType	0	<input type="text" value="Unsigned Integer"/>		0,1,2
DataWidth	32	<input type="text" value="32"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian (SwapBytesAr"/>		0,1,2,3
BitFieldWidth	32	<input type="text" value="0"/>	bits	[0-64]
BitFieldPosition	0	<input type="text" value="0"/>	bit	[0-63]
Divisor	1	<input type="text" value="1"/>		
InitialValueAccess	0	<input type="text" value="Read"/>		0,1
InitialValue	-	<input type="text" value="0"/>		
Value	500000			
RawValue	547424000			
IsValueValid	1		bool	
ErrorCode	0			

Auto odświeżanie

Odczyt 32 - bitowych wartości całkowitych rejestru wejściowego (`Read Input Registers`, `FunctionCode = 04`):

- `RegisterType` : Input Registers;
- `DataType` : Unsigned Integer;
- `DataWidth` : 32;
- `Endianness` : Big Endian;
- `BitFieldWidth` : wartość domyślna;
- `BitFieldPosition` : wartość domyślna.

Zapis 32 - bitowych wartości całkowitych w rejestrze pamiętającym (`Preset/Write Multiple Holding Registers`, `FunctionCode = 16`):

- `RegisterType` : Holding Registers;
- `DataType` : Unsigned Integer;
- `DataWidth` : 32;
- `Endianness` : Big Endian;
- `BitFieldWidth` : wartość domyślna;
- `BitFieldPosition` : wartość domyślna.

D. Pola bitowe w rejestrach 32 - bitowych

Odczyt pół bitowych w 32 - bitowym rejestrze pamiętającym (`Read Holding Registers`, `FunctionCode = 03`):

- `RegisterType` : Holding Registers;
- `DataType` : Unsigned Integer;
- `DataWidth` : 32;
- `Endianness` : wartość domyślna;
- `BitFieldWidth` : 0 - 32 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = `DataWidth`);
- `BitFieldPosition` : 0 - 31 (pozycja pierwszego interesującego bitu).

Uwaga!

Zakres liczbowy cechy `BitFieldWidth` zależy od ustawienia wartości cechy `DataWidth`.

Dla `DataWidth = 32` zakres cechy `BitFieldWidth` wynosi [0 - 32].

W chwili ustawienia cechy `BitFieldWidth` na wartość równą 0 i wysłaniu konfiguracji cecha przyjmuje wartość maksymalną dla aktualnie ustawionej wartości cechy `DataWidth`.

Uwaga!

Zakres liczbowy cechy `BitFieldPosition` zależy od ustawienia wartości cechy `DataWidth`.

Dla `DataWidth = 32` zakres cechy `BitFieldPosition` wynosi [0 - 31].

Uwaga!

Cechy `BitFieldWidth` i `BitFieldPosition` są uzależnione od cechy `DataWidth`.

Działają one wg zależności: `BitFieldWidth + BitFieldPosition <= DataWidth`

Na przykład:

W chwili ustawienia `DataWidth` i `BitFieldWidth` = 32 oraz `BitFieldPosition` = 15 cecha `BitFieldWidth` automatycznie przyjmie wartość = 17.

Dla `BitFieldWidth` = 0 cecha `BitFieldPosition` przyjmuje zawsze 0.

Właściwości obiektu

Nazwa: Typ:
Id:

Sterowanie Zdarzenia Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
TransmissionSpeed	19200	<input type="text" value="19200"/>	bps	1200,2400,4800,9600,19200,3840
Parity	0	<input type="text" value="None"/>		0,1,2
StopBits	0	<input type="text" value="1"/>	bits	0,1,2
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
ResponseTimeout	300	<input type="text" value="300"/>	ms	[25-6400]
RefreshInterval	1000	<input type="text" value="1000"/>	ms	[0-300000]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	<input type="text" value="Rejestry pamiętające (ho"/>		0,1,2,3
AlwaysWriteMultiple	0	<input type="text" value="No"/>		0,1
InputOutputCount	0	<input type="text" value="1"/>		[1-64]
DataType	0	<input type="text" value="Unsigned Integer"/>		0,1,2
DataWidth	32	<input type="text" value="32"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian (SwapBytesAr"/>		0,1,2,3
BitFieldWidth	17	<input type="text" value="17"/>	bits	[0-64]
BitFieldPosition	15	<input type="text" value="15"/>	bit	[0-63]
Divisor	1	<input type="text" value="1"/>		
InitialValueAccess	0	<input type="text" value="Read"/>		0,1
InitialValue	-	<input type="text" value="0"/>		
Value	15			
RawValue	547424000			
IsValueValid	1		bool	
ErrorCode	0			

Auto odświeżanie

Odczyt pól bitowych w 32 - bitowym rejestrze wejściowym (`Read Input Registers`, `FunctionCode` = 04):

- `RegisterType`: Input Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 32;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: 0 - 32 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = `DataWidth`);
- `BitFieldPosition`: 0 - 31 (pozycja pierwszego interesującego bitu).

Zapis pól bitowych w 16-bitowym rejestrze pamiętającym (`Preset/Write Single Holding Register`, `FunctionCode = 06`):

- `RegisterType` : Holding Registers;
- `DataType` : Unsigned Integer;
- `DataWidth` : 32;
- `Endianness` : wartość domyślna;
- `BitFieldWidth` : 0 - 32 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = `DataWidth`);
- `BitFieldPosition` : 0 - 31 (pozycja pierwszego interesującego bitu).

E. 32 - bitowe wartości zmiennoprzecinkowe rejestrów

Odczyt 32 - bitowych wartości zmiennoprzecinkowych rejestru pamiętającego (`Read Holding Registers`, `FunctionCode = 03`):

- `RegisterType` : Holding Registers;
- `DataType` : Floating-point;
- `DataWidth` : 32;
- `Endianness` : Big Endian;

Dla `DataType = Floating-point` cechy `BitFieldWidth`, `BitFieldPosition` oraz `Divisor` są nieaktywne i przyjmują zawsze wartość 0!



Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
TransmissionSpeed	19200	<input type="text" value="19200"/>	bps	1200,2400,4800,9600,19200,38400
Parity	0	<input type="text" value="None"/>		0,1,2
StopBits	0	<input type="text" value="1"/>	bits	0,1,2
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
ResponseTimeout	300	<input type="text" value="300"/>	ms	[25-6400]
RefreshInterval	1000	<input type="text" value="1000"/>	ms	[0-300000]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	<input type="text" value="Rejestry pamiętające (ho"/>		0,1,2,3
AlwaysWriteMultiple	0	<input type="text" value="No"/>		0,1
InputOutputCount	0	<input type="text" value="1"/>		[1-64]
DataType	2	<input type="text" value="Floating-point"/>		0,1,2
DataWidth	32	<input type="text" value="32"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian (SwapBytesAr"/>		0,1,2,3
BitFieldWidth	0	<input type="text" value="0"/>	bits	[0-64]
BitFieldPosition	0	<input type="text" value="0"/>	bit	[0-63]
Divisor	0	<input type="text" value="1"/>		
InitialValueAccess	0	<input type="text" value="Read"/>		0,1
InitialValue	-	<input type="text" value="0"/>		
Value	100.00			
RawValue	51266			
IsValueValid	1		bool	
ErrorCode	0			

Auto odświeżanie 
 Odśwież

Odczyt 32 - bitowych wartości zmiennoprzecinkowych rejestru wejściowego (`Read Input Registers`, `FunctionCode = 04`):

- `RegisterType` : Input Registers;
- `DataType` : Floating-point;
- `DataWidth` : 32;
- `Endianness` : Big Endian.

Zapis 32 - bitowych wartości zmiennoprzecinkowych w rejestrze pamiętającym (`Preset/Write Multiple Holding Registers`, `FunctionCode = 16`):

- `RegisterType` : Holding Registers;
- `DataType` : Floating-point;
- `DataWidth` : 32;
- `Endianness` : Big Endian.

F. 64-bitowe wartości całkowite rejestrów

Odczyt 64-bitowych rejestrów pamiętających (`Read Holding Registers` , `FunctionCode = 03`):

- `RegisterType` : Holding Registers;
- `DataType` : Unsigned Integer;
- `DataWidth` : 64
- `Endianness` : wartość domyślna;
- `BitFieldWidth` : wartość domyślna;
- `BitFieldPosition` : wartość domyślna.

Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie Zdarzenia Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
TransmissionSpeed	19200	<input type="text" value="19200"/>	bps	1200,2400,4800,9600,19200,38400
Parity	0	<input type="text" value="None"/>		0,1,2
StopBits	0	<input type="text" value="1"/>	bits	0,1,2
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
ResponseTimeout	300	<input type="text" value="300"/>	ms	[25-6400]
RefreshInterval	1000	<input type="text" value="1000"/>	ms	[0-300000]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	<input type="text" value="Rejestry pamiętające (holding registers)"/>		0,1,2,3
AlwaysWriteMultiple	0	<input type="text" value="No"/>		0,1
InputOutputCount	0	<input type="text" value="1"/>		[1-64]
DataType	0	<input type="text" value="Unsigned Integer"/>		0,1,2
DataWidth	64	<input type="text" value="64"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian (SwapBytesAr)"/>		0,1,2,3
BitFieldWidth	64	<input type="text" value="0"/>	bits	[0-64]
BitFieldPosition	0	<input type="text" value="0"/>	bit	[0-63]
Divisor	1	<input type="text" value="1"/>		
InitialValueAccess	0	<input type="text" value="Read"/>		0,1
InitialValue	-	<input type="text" value="0"/>		
Value	100200300400			
RawValue	8088297066641489920			
IsValueValid	1		bool	
ErrorCode	0			

Auto odświeżanie

Odczyt 64-bitowych rejestrów wejściowych (`Read Input Registers` , `FunctionCode = 04`):

- `RegisterType` : Input Registers;
- `DataType` : Unsigned Integer;
- `DataWidth` : 64
- `Endianness` : wartość domyślna;
- `BitFieldWidth` : wartość domyślna;
- `BitFieldPosition` : wartość domyślna.

Zapis 64-bitowych rejestrów pamiętających (`Preset / Write Single Holding Register`, `FunctionCode = 06`):

- `RegisterType`: Holding Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 64;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: wartość domyślna;
- `BitFieldPosition`: wartość domyślna.

G. Pola bitowe w rejestrach 64-bitowych

Odczyt pól bitowych w 64-bitowym rejestrze pamiętającym (`Read Holding Registers`, `FunctionCode = 03`):

- `RegisterType`: Holding Registers;
- `DataType`: Unsigned Integer;
- `DataWidth`: 64;
- `Endianness`: wartość domyślna;
- `BitFieldWidth`: 0 - 64 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = `DataWidth`);
- `BitFieldPosition`: 0 - 63 (pozycja pierwszego interesującego bitu).

Uwaga!

Zakres liczbowy cechy `BitFieldWidth` zależy od ustawienia wartości cechy `DataWidth`.

Dla `DataWidth = 64` zakres cechy `BitFieldWidth` wynosi [0 - 64].

W chwili ustawienia cechy `BitFieldWidth` na wartość równą 0 i wysłaniu konfiguracji cecha przyjmuje wartość maksymalną dla aktualnie ustawionej wartości cechy `DataWidth`.

Uwaga!

Zakres liczbowy cechy `BitFieldPosition` zależy od ustawienia wartości cechy `DataWidth`.

Dla `DataWidth = 64` zakres cechy `BitFieldPosition` wynosi [0 - 63].

Uwaga!

Cechy `BitFieldWidth` i `BitFieldPosition` są uzależnione od cechy `DataWidth`.

Działają one wg zależności: $\text{BitFieldWidth} + \text{BitFieldPosition} \leq \text{DataWidth}$

Na przykład:

W chwili ustawienia `DataWidth = 60` oraz `BitFieldPosition = 15` cecha `BitFieldWidth` automatycznie przyjmie wartość = 49

Dla `BitFieldWidth = 0` cecha `BitFieldPosition` przyjmuje zawsze 0.

Właściwości obiektu
✕

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
TransmissionSpeed	19200	<input type="text" value="19200"/>	bps	1200,2400,4800,9600,19200,38400
Parity	0	<input type="text" value="None"/>		0,1,2
StopBits	0	<input type="text" value="1"/>	bits	0,1,2
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
ResponseTimeout	300	<input type="text" value="300"/>	ms	[25-6400]
RefreshInterval	1000	<input type="text" value="1000"/>	ms	[0-300000]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	<input type="text" value="Rejestry pamiętające (ho"/>		0,1,2,3
AlwaysWriteMultiple	0	<input type="text" value="No"/>		0,1
InputOutputCount	0	<input type="text" value="1"/>		[1-64]
DataType	0	<input type="text" value="Unsigned Integer"/>		0,1,2
DataWidth	64	<input type="text" value="64"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian (SwapBytesAr"/>		0,1,2,3
BitFieldWidth	49	<input type="text" value="60"/>	bits	[0-64]
BitFieldPosition	15	<input type="text" value="15"/>	bit	[0-63]
Divisor	1	<input type="text" value="1"/>		
InitialValueAccess	0	<input type="text" value="Read"/>		0,1
InitialValue	-	<input type="text" value="0"/>		
Value	3057870			
RawValue	8088297066641489920			
IsValueValid	1		bool	
ErrorCode	0			

Auto odświeżanie

Odczyt pól bitowych w 64-bitowym rejestrze wejściowym (`Read Input Registers`, `FunctionCode = 04`):

- `RegisterType` : Input Registers;
- `DataType` : Unsigned Integer;
- `DataWidth` : 64;
- `Endianness` : wartość domyślna;
- `BitFieldWidth` : 0 - 64 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = `DataWidth`);
- `BitFieldPosition` : 0 - 63 (pozycja pierwszego interesującego bitu).

Zapis pól bitowych w 16-bitowym rejestrze pamiętającym (`Preset/Write Single Holding Register`, `FunctionCode = 06`):

- `RegisterType` : Holding Registers;
- `DataType` : Unsigned Integer;
- `DataWidth` : 64;

- `Endianness` : wartość domyślna;
- `BitFieldWidth` : 0 - 64 (ilość odczytywanych kolejno bitów; dla wartości 0 przyjmowana jest automatycznie szerokość maksymalna = `DataWidth`);
- `BitFieldPosition` : 0 - 63 (pozycja pierwszego interesującego bitu).

H. 64-bitowe wartości zmiennoprzecinkowe rejestrów

Odczyt 64-bitowych wartości zmiennoprzecinkowych rejestru pamiętającego (`Read Holding Registers` , `FunctionCode = 03`):

- `RegisterType` : Holding Registers;
- `DataType` : Floating-point;
- `DataWidth` : 64;
- `Endianness` : Big Endian.

Dla `DataType = Floating-point` cechy `BitFieldWidth` , `BitFieldPosition` oraz `Divisor` są nieaktywne i przyjmują zawsze wartość 0!

↑
×

Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
TransmissionSpeed	19200	<input type="text" value="19200"/>	bps	1200,2400,4800,9600,19200,3840
Parity	0	<input type="text" value="None"/>		0,1,2
StopBits	0	<input type="text" value="1"/>	bits	0,1,2
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
ResponseTimeout	300	<input type="text" value="300"/>	ms	[25-6400]
RefreshInterval	1000	<input type="text" value="1000"/>	ms	[0-300000]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	<input type="text" value="Rejestry pamiętające (ho"/>		0,1,2,3
AlwaysWriteMultiple	0	<input type="text" value="No"/>		0,1
InputOutputCount	0	<input type="text" value="1"/>		[1-64]
DataType	2	<input type="text" value="Floating-point"/>		0,1,2
DataWidth	64	<input type="text" value="64"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian (SwapBytesAr"/>		0,1,2,3
BitFieldWidth	0	<input type="text" value="0"/>	bits	[0-64]
BitFieldPosition	0	<input type="text" value="0"/>	bit	[0-63]
Divisor	0	<input type="text" value="1"/>		
InitialValueAccess	0	<input type="text" value="Read"/>		0,1
InitialValue	-	<input type="text" value="500100.78"/>		
Value	500100.78			
RawValue	17028594105380642369			
IsValueValid	1		bool	
ErrorCode	0			

Auto odświeżanie

Odczyt 64-bitowych wartości zmiennoprzecinkowych rejestru wejściowego (`Read Input Registers`, `FunctionCode = 04`):

- `RegisterType`: Input Registers;
- `DataType`: Floating-point;
- `DataWidth`: 64;
- `Endianness`: Big Endian.

Zapis 64-bitowych wartości zmiennoprzecinkowych w rejestrze pamiętającym (`Preset/Write Multiple Holding Registers`, `FunctionCode = 16`):

- `RegisterType`: Holding Registers;
- `DataType`: Floating-point;
- `DataWidth`: 64;
- `Endianness`: Big Endian.

J. Dyskretne wejścia / wyjścia

Odczyt dyskretnych wyjść / wejść bitowych (`Read Coils`, `FunctionCode = 01`):

- `RegisterType`: Discrete outputs / coils;
- `InputOutputCount`: 1 - 64 (liczba dyskretnych wejść / wyjść podlegających operacji odczytu / zapisu).

Dla typów rejestrów `Discrete outputs / coils` i `Discrete Inputs` cechy `DataType`, `DataWidth`, `Endianness`, `BitFieldWidth`, `BitFieldPosition`, `Divisor` są nieaktywne i przyjmują wartość 0!

Właściwości obiektu
✕

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
TransmissionSpeed	19200	<input type="text" value="19200"/>	bps	1200,2400,4800,9600,19200,38400
Parity	0	<input type="text" value="None"/>		0,1,2
StopBits	0	<input type="text" value="1"/>	bits	0,1,2
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
ResponseTimeout	300	<input type="text" value="300"/>	ms	[25-6400]
RefreshInterval	1000	<input type="text" value="1000"/>	ms	[0-300000]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	0	<input type="text" value="Wyjścia dyskretne (coil)"/>		0,1,2,3
AlwaysWriteMultiple	0	<input type="text" value="No"/>		0,1
InputOutputCount	8	<input type="text" value="8"/>		[1-64]
DataType	0	<input type="text" value="Floating-point"/>		0,1,2
DataWidth	0	<input type="text" value="64"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian (SwapBytesAr)"/>		0,1,2,3
BitFieldWidth	0	<input type="text" value="0"/>	bits	[0-64]
BitFieldPosition	0	<input type="text" value="0"/>	bit	[0-63]
Divisor	0	<input type="text" value="1"/>		
InitialValueAccess	0	<input type="text" value="Read"/>		0,1
InitialValue	-	<input type="text" value="0"/>		
Value	15			
RawValue	15			
IsValueValid	1		bool	
ErrorCode	0			

Auto odświeżanie

Odczyt dyskretnych wejść dwustanowych (, *FunctionCode = 02*):

- : Discrete Inputs;
- : 1 - 64 (liczba dyskretnych wejść / wyjść podlegających operacji odczytu / zapisu).

Zapis dyskretnych wyjść/wejść bitowych (, *FunctionCode = 05*)

- : Discrete outputs / coils;
- : No
- : 1 (liczba dyskretnych wejść / wyjść podlegających operacji odczytu / zapisu).

Zapis dyskretnych wyjść/wejść bitowych (, *FunctionCode = 15*)

- : Discrete outputs / coils;
- : No
- : 2 - 64 (liczba dyskretnych wejść / wyjść podlegających operacji odczytu / zapisu).

Uwaga!

W przypadku zapisu wartości do pojedynczego wejścia dwustanowego `InputOutputCount = 1` możliwe jest użycie funkcji 15 (`Write Single Coil`) zamiast funkcji 5 (`Write Multiple Coils`) w przypadku kiedy urządzenie Slave akceptuje jedynie zapis tego typu. W tym celu należy użyć cechy `AlwaysWriteMultiple` i ustawić ją na wartość 1.

Zapis dyskretnych wyjść/wejść bitowych (`Write Multiple Coils`, `FunctionCode = 15`):

- `RegisterType`: Discrete outputs / coils;
- `AlwaysWriteMultiple`: Yes
- `InputOutputCount`: 1 (liczba dyskretnych wejść / wyjść podlegających operacji odczytu / zapisu).

4.2. Obiekty ModbusSlaveRTU oraz ModbusServer

Obiekty `ModbusSlaveRTU` / `ModbusServer` służą do przechowywania danych za pomocą rejestrów dostępnych w protokole MODBUS. Parametry rejestrów w przypadku obiektów nie różnią się w obszarze konfiguracji - jedyna różnica polega na sposobie komunikacji z docelowym urządzeniem. Poniżej przedstawiono przykładowe sposoby przechowywania wartości w rejestrach.

A. 16-bitowe wartości całkowite rejestrów

Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie Zdarzenia Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	Rejestry pamiętające (ho <input type="text" value="..."/>)		0,1,2,3
DataType	0	Unsigned Integer <input type="text" value="..."/>		0,1,2
DataWidth	16	<input type="text" value="16"/> bits		16,32,48,64
Endianness	0	Big Endian (SwapBytesAr <input type="text" value="..."/>)		0,1,2,3
InitialValue	-	<input type="text" value="1996"/>		
Value	1996			
RawValue	52231			

Auto odświeżanie

Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
Port	502	<input type="text" value="502"/>		
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	<input type="text" value="Rejestry pamiętające (ho"/>		0,1,2,3
DataType	0	<input type="text" value="Unsigned Integer"/>		0,1,2
DataWidth	16	<input type="text" value="16"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian (SwapBytesAr"/>		0,1,2,3
InitialValue	-	<input type="text" value="1996"/>		
Value	1996			
RawValue	52231			

Auto odświeżanie

B. 16-bitowe wartości całkowite ujemne rejestrów

Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	<input type="text" value="Rejestry pamiętające (ho"/>		0,1,2,3
DataType	1	<input type="text" value="Signed Integer"/>		0,1,2
DataWidth	16	<input type="text" value="16"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian (SwapBytesAr"/>		0,1,2,3
InitialValue	-	<input type="text" value="-1996"/>		
Value	-1996			
RawValue	13560			

Auto odświeżanie

C. 32 - bitowe wartości całkowite rejestrów

Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	Rejestry pamiętające (ho		0,1,2,3
DataType	0	Unsigned Integer		0,1,2
DataWidth	32	<input type="text" value="32"/>	bits	16,32,48,64
Endianness	0	Big Endian (SwapBytesAr		0,1,2,3
InitialValue	-	<input type="text" value="100200300"/>		
Value	100200300			
RawValue	1827665925			

Auto odświeżanie

D. 32 - bitowe wartości całkowite ujemne rejestrów

Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	Rejestry pamiętające (ho		0,1,2,3
DataType	1	Signed Integer		0,1,2
DataWidth	32	<input type="text" value="32"/>	bits	16,32,48,64
Endianness	0	Big Endian (SwapBytesAr		0,1,2,3
InitialValue	-	<input type="text" value="-100200300"/>		
Value	-100200300			
RawValue	2484078586			

Auto odświeżanie

E. 32 - bitowe wartości zmiennoprzecinkowe

Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	2	Rejestry pamiętające (ho		0,1,2,3
DataType	2	Floating-point		0,1,2
DataWidth	32	<input type="text" value="32"/>	bits	16,32,48,64
Endianness	0	Big Endian (SwapBytesAr		0,1,2,3
InitialValue	-	<input type="text" value="100.78"/>		
Value	100.78			
RawValue	1552927042			

Auto odświeżanie

F. Wyjścia dyskretne (coil) i Wejścia dyskretne

Właściwości obiektu

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	0	Wyjścia dyskretne (coil)		0,1,2,3
DataType	0	Signed Integer		0,1,2
DataWidth	0	<input type="text" value="32"/>	bits	16,32,48,64
Endianness	0	Big Endian (SwapBytesAr		0,1,2,3
InitialValue	-	<input type="text" value="1"/>		
Value	1			
RawValue	1			

Auto odświeżanie

Właściwości obiektu
×

Nazwa: Typ:

Id:

Sterowanie
 Zdarzenia
 Cechy wbudowane

Nazwa cechy	Aktualna wartość	Wartość początkowa	Jednostka	Zakres
DeviceAddress	1	<input type="text" value="1"/>		[0-255]
RegisterAddress	1	<input type="text" value="1"/>		[0-65535]
RegisterType	1	<input type="text" value="Wejścia dyskretne"/>		0,1,2,3
DataType	0	<input type="text" value="Signed Integer"/>		0,1,2
DataWidth	0	<input type="text" value="32"/>	bits	16,32,48,64
Endianness	0	<input type="text" value="Big Endian (SwapBytesAr"/>		0,1,2,3
InitialValue	-	<input type="text" value="1"/>		
Value	1			
RawValue	1			

Auto odświeżanie

Uwaga!

Jeden obiekt `ModbusSlaveRTU` dla typu rejestru `Wejścia dyskretne` i `Wyjścia dyskretne` odpowiada jednemu bitowi danych. Każdy kolejny utworzony obiekt to kolejny bit.

5. Przywracanie ustawień fabrycznych - *Hard Reset*

Uruchomienie funkcji *Hard Reset* na module GATE Modbus powoduje:

- Usunięcie zapisanej konfiguracji;
- Sformatowanie partycji pamięci flash;
- Usunięcie wszystkich utworzonych obiektów LUA;
- Utratę komunikacji pomiędzy OM / HM a modulem Gate.

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

- Odłączyć zasilanie od modułu Gate;
- Nacisnąć i przytrzymać przycisk *Reset* na module (przycisk znajduje się pod dolną zaślepką modułu);
- Podłączyć zasilanie do modułu Gate;
- Trzymać wciśnięty przycisk *Reset* przez co najmniej 10 sekund - podczas resetu dioda zielona będzie świecić światłem ciągłym. Prawidłowe wykonanie resetu zostanie potwierdzone 3-krotnym mrugnięciem diody zielonej.
- Po upływie 10 sekund zwolnić przycisk *Reset*
- Odczekać około 60 sekund aż do momentu, gdy na module diody - zielona oraz czerwona - będą mrugać naprzemiennie (tryb *Emergency*)

Po wykonaniu procedury na module zostanie wyczyszczona konfiguracja, natomiast sam moduł przestanie być widoczny (brak odpowiedzi na *Keep-Alive*) w projekcie z poziomu Object Managera. Aby ponownie przywrócić moduł, należy wykonać CLU Discovery a następnie wysłać konfigurację.

6. Parametry konfiguracyjne

Uwaga!

Opisana funkcjonalność oraz integracja jest dostępna dla **GRENTON GATE MODBUS MASTER, DIN, Eth (INT-201 - E-01)** posiadający **firmware 1.4.1 - 2334 lub wyższy!**

A. Obiekt GATE

CECHY

Nazwa	Opis
Uptime	Czas pracy urządzenia od ostatniego resetu (w sekundach)
ClientReportInterval	Okres raportowania o zmianach cech
Date	Zwraca aktualną datę
Time	Zwraca aktualny czas (hh:mm:ss)
LocalTime	Zwraca aktualny znacznik czasu
TimeZone	Strefa czasowa
UnixTime	Zwraca aktualny uniksowy znacznik czasu
FirmwareVersion	Wersja oprogramowania Gate
UseCloud	Określa czy GATE łączy się do chmury
CloudConnection	Określa status połączenia GATE z chmurą
NTPTimeout	Czas oczekiwania na odpowiedź z serwera NTP
UseNTP	Określa czy GATE używa NTP
PrimaryDNS	Preferowany serwer DNS
SecondaryDNS	Alternatywny serwer DNS
TelnetLogLevel	Określa poziom logowania
TelnetBusLogLevel	Określa poziom logowania Modbus
ModbusMasterFrameSpace	Określa dodatkową przerwę pomiędzy ramkami Modbus wyrażoną w znakach
OverloadDetection	Określa, czy Gate powinien zgłaszać przeciążenie procesora używając czerwonej diody
ResetReason	Określa przyczynę restartu urządzenia: 0 - włączenie zasilania 2 - przeładowanie konfiguracji 3 - wyjątek systemowy

METODY

Nazwa	Opis
SetDateTime	Ustawia datę i czas
SetClientReportInterval	Ustawia okres raportowania o zmianach cech
SetPrimaryDNS	Ustawia cechę PrimaryDNS
SetSecondaryDNS	Ustawia cechę SecondaryDNS
SetTelnetLogLevel	Określa poziom logowania
SetTelnetBusLogLevel	Określa poziom logowania Modbus

ZDARZENIA

Nazwa	Opis
OnInit	Zdarzenie wywoływane jednorazowo w momencie inicjalizacji urządzenia

B. Obiekt ModbusRTU

Uwaga!

Opisany obiekt wirtualny jest dostępny dla **GRENTON GATE MODBUS MASTER, DIN, Eth (INT-201 - E-01)** posiadający **firmware 1.4.1 - 2334 lub wyższy!**

CECHY

Nazwa	Opis
TransmissionSpeed	Prędkość transmisji
Parity	Bit parzystości: 0 - None 1 - Odd 2 - Even
StopBits	Bity stopu: 0 - 1 bit stopu 1 - 1.5 bitu stopu 2 - 2 bity stopu
DeviceAddress	Adres urządzenia Modbus
ResponseTimeout	Czas na odpowiedź na ramkę w krokach 25ms
RefreshPeriod	Minimalny okres odświeżania w krokach 5ms. Wartość 0 wyłącza automatyczne odświeżanie
RegisterAddress	Adres obsługiwanego rejestru
RegisterType	Typ rejestru Modbus: 0 - wyjścia dwustanowe (coils) - funkcja Modbus: 5 (zapis pojedynczego wyjścia), 15 (zapis wielu wyjść) lub 1 (odczyt stanu wyjść) 1 - wejścia dwustanowe - funkcja Modbus: 2 2 - rejestry pamiętające - funkcja Modbus: 6 (zapis pojedynczego rejestru), 16 (zapis wielu rejestrów) lub 3 (odczyt rejestrów) 3 - rejestry wejściowe - funkcja Modbus: 4
AlwaysWriteMultiple	Zawsze używaj funkcji 15 lub 16 do zapisu wartości
InputOutputCount	Określa liczbę dyskretnych wejść / wyjść podlegających operacji odczytu / zapisu
DataType	Typ wartości: 0 - Liczba całkowita, stałoprzecinkowa lub pole bitowe bez bitu znaku 1 - Liczba całkowita, stałoprzecinkowa lub pole bitowe z bitem znaku 2 - Liczba zmiennoprzecinkowa
DataWidth	Szerokość danych (od 1 do 4 rejestrów 16 bitowych)
Endianness	Kolejność bajtów: 0 - kolejność słów: Big Endian; kolejność bajtów w słowie: Big Endian (SwapBytesAndWords) 1 - kolejność słów: Little Endian; kolejność bajtów w słowie: Big Endian (SwapBytes) 2 - kolejność słów: Big Endian; kolejność bajtów w słowie: Little Endian (SwapWords) 3 - kolejność słów: Little Endian; kolejność bajtów w słowie: Little Endian (NoSwap)

Nazwa	Opis
<code>BitFieldWidth</code>	Liczba bitów pola bitowego. Suma <code>BitFieldWidth</code> i <code>BitFieldPosition</code> powinna być \leq <code>DataWidth</code> ; 0 oznacza brak pola bitowego (pełna szerokość danych = <code>DataWidth</code>)
<code>BitFieldPosition</code>	Pozycja najmłodszego bitu pola bitowego. Suma <code>BitFieldWidth</code> i <code>BitFieldPosition</code> powinna być \leq <code>DataWidth</code> .
<code>Divisor</code>	Dzielnik
<code>InitialValueAccess</code>	Metoda pierwszego dostępu do wartości Value: 0 - wartość początkowa Value jest odczytywana z urządzenia 1 - wartość początkowa Value jest zapisywana do urządzenia
<code>InitialValue</code>	Określa wartość początkową
<code>Value</code>	Zwraca ostatnio odczytaną wartość
<code>RawValue</code>	Nieprzeskalowana wartość rejestru
<code>IsValid</code>	Określa, czy wartość jest zgodna ze stanem obiektu
<code>ErrorCode</code>	Kod błędu: 1 - niedozwolona funkcja 2 - niedozwolony numer rejestru 3 - niedozwolona wartość danej 4 - uszkodzenie w przyłączonym urządzeniu 5 - potwierdzenie pozytywne 6 - brak gotowości, komunikat usunięty 7 - potwierdzenie negatywne 8 - błąd parzystości pamięci 10 - ścieżka bramy niedostępna 11 - brak odpowiedzi urządzenia docelowego 0 - poprawny odczyt/zapis rejestru -2 - przekroczenie czasu odpowiedzi -3 - błąd ramki (problem ze zdekodowaniem odpowiedzi) -4 - nieoczekiwany rozmiar odpowiedzi -5 - nieoczekiwany kod odpowiedzi -6 - nieprawidłowy stan obiektu -7 - błąd połączenia

METODY

Nazwa	Opis
<code>SetTransmissionSpeed</code>	Ustawia prędkość transmisji
<code>SetParity</code>	Ustawia sprawdzanie parzystości
<code>SetStopBits</code>	Ustawia liczbę bitów stopu
<code>SetDeviceAddress</code>	Ustawia adres urządzenia Modbus
<code>SetResponseTimeout</code>	Ustawia czas przekroczenia odpowiedzi w krokach 25ms
<code>SetRefreshPeriod</code>	Ustawia okres odświeżania w krokach 5ms - wartość 0 wyłącza automatyczne odświeżanie
<code>SetRegisterAddress</code>	Ustawia adres obsługiwanego rejestru
<code>SetRegisterType</code>	Ustawia typ rejestru Modbus
<code>SetAlwaysWriteMultiple</code>	Zawsze używaj funkcji 15 lub 16 do zapisu wartości
<code>SetInputOutputCount</code>	Ustawia liczbę dyskretnych wejść / wyjść podlegających operacji odczytu / zapisu
<code>SetDataType</code>	Ustawia typ zmiennej
<code>SetDataWidth</code>	Ustawia szerokość danych
<code>SetEndianness</code>	Ustawia typ kolejności bajtów
<code>SetBitFieldWidth</code>	Ustawia liczbę bitów pola - 0 oznacza brak pola bitowego (pełna szerokość danych <code>DataWidth</code>)
<code>SetBitFieldPosition</code>	Ustawia pozycję najmłodszego bitu pola
<code>SetDivisor</code>	Ustawia dzielnik
<code>ReadValue</code>	Rozpoczyna odczyt wartości z urządzenia - czeka na zakończenie odczytu w przypadku braku <code>OnValueRead</code>
<code>WriteValue</code>	Zapisuje nową wartość do urządzenia

ZDARZENIA

Nazwa	Opis
<code>OnValueChange</code>	Zdarzenie wywoływane w przypadku zmiany wartości cechy Value przez Modbus
<code>OnValueRead</code>	Zdarzenie wywoływane po zakończeniu odczytu rozpoczętego przez metodę <code>ReadValue</code>
<code>OnValueWritten</code>	Zdarzenie wywołane po zakończeniu zapisu rozpoczętego przez metodę <code>WriteValue</code>
<code>OnError</code>	Zdarzenie wywoływane gdy urządzenie zgłasza błąd

C. Obiekt ModbusSlaveConfigRTU

Uwaga!

Opisany obiekt wirtualny jest dostępny dla **GRENTON GATE MODBUS MASTER, DIN, Eth (INT-201 - E-01)** posiadający **firmware 1.4.1 - 2334 lub wyższy!**

CECHY

Nazwa	Opis
TransmissionSpeed	Prędkość transmisji
Parity	Bit parzystości: 0 - None 1 - Odd 2 - Even
StopBits	Bity stopu: 0 - 1 bit stopu 1 - 1.5 bitu stopu 2 - 2 bity stopu

D. Obiekt ModbusSlaveRTU**Uwaga!**

Opisany obiekt wirtualny jest dostępny dla **GRENTON GATE MODBUS MASTER, DIN, Eth (INT-201 - E-01)** posiadający **firmware 1.4.1 - 2334 lub wyższy!**

CECHY

Nazwa	Opis
DeviceAddress	Adres urządzenia Modbus
RegisterAddress	Adres obsługiwane go rejestru
RegisterType	Typ rejestru Modbus: 0 - wyjścia dwustanowe (coils) - funkcja Modbus: 5 (zapis pojedynczego wyjścia), 15 (zapis wielu wyjść) lub 1 (odczyt stanu wyjść) 1 - wejścia dwustanowe - funkcja Modbus: 2 2 - rejestry pamiętające - funkcja Modbus: 6 (zapis pojedynczego rejestru), 16 (zapis wielu rejestrów) lub 3 (odczyt rejestrów) 3 - rejestry wejściowe - funkcja Modbus: 4
DataType	Typ wartości: 0 - Liczba całkowita, stałoprzecinkowa lub pole bitowe bez bitu znaku 1 - Liczba całkowita, stałoprzecinkowa lub pole bitowe z bitem znaku 2 - Liczba zmiennoprzecinkowa
DataWidth	Szerokość danych (od 1 do 4 rejestrów 16 bitowych)
Endianness	Kolejność bajtów: 0 - kolejność słów: Big Endian; kolejność bajtów w słowie: Big Endian (SwapBytesAndWords) 1 - kolejność słów: Little Endian; kolejność bajtów w słowie: Big Endian (SwapBytes) 2 - kolejność słów: Big Endian; kolejność bajtów w słowie: Little Endian (SwapWords) 3 - kolejność słów: Little Endian; kolejność bajtów w słowie: Little Endian (NoSwap)
InitialValue	Określa wartość początkową
Value	Wartość rejestru
RawValue	Nieprzeskalowana wartość rejestru

METODY

Nazwa	Opis
SetValue	Ustawia wartość rejestru

ZDARZENIA

Nazwa	Opis
OnValueChange	Zdarzenie wywoływane w przypadku zmiany wartości cechy Value przez Modbus

E. Obiekt ModbusClient

Uwaga!

Opisany obiekt wirtualny jest dostępny dla **GRENTON GATE MODBUS MASTER, DIN, Eth (INT-201 - E-01)** posiadający **firmware 1.4.1 - 2334 lub wyższy!**

CECHY

Nazwa	Opis
SocketAddress	Adres IP urządzenia
DeviceAddress	Adres urządzenia Modbus
ResponseTimeout	Czas na odpowiedź w krokach 25ms
RefreshPeriod	Minimalny okres odświeżania w krokach 5ms. Wartość 0 wyłącza automatyczne odświeżanie
RegisterAddress	Adres obsługiwanego rejestru
RegisterType	<p>Typ rejestru Modbus:</p> <p>0 - wyjścia dwustanowe (coils) - funkcja Modbus: 5 (zapis pojedynczego wyjścia), 15 (zapis wielu wyjść) lub 1 (odczyt stanu wyjść)</p> <p>1 - wejścia dwustanowe - funkcja Modbus: 2</p> <p>2 - rejestry pamiętające - funkcja Modbus: 6 (zapis pojedynczego rejestru), 16 (zapis wielu rejestrów) lub 3 (odczyt rejestrów)</p> <p>3 - rejestry wejściowe - funkcja Modbus: 4</p>
AlwaysWriteMultiple	Zawsze używaj funkcji 15 lub 16 do zapisu wartości
InputOutputCount	Określa liczbę dyskretnych wejść / wyjść podlegających operacji odczytu / zapisu
DataType	<p>Typ wartości:</p> <p>0 - Liczba całkowita, stałoprzecinkowa lub pole bitowe bez bitu znaku</p> <p>1 - Liczba całkowita, stałoprzecinkowa lub pole bitowe z bitem znaku</p> <p>2 - Liczba zmiennoprzecinkowa</p>
DataWidth	Szerokość danych (od 1 do 4 rejestrów 16 bitowych)
Endianness	<p>Kolejność bajtów:</p> <p>0 - kolejność słów: Big Endian; kolejność bajtów w słowie: Big Endian (SwapBytesAndWords)</p> <p>1 - kolejność słów: Little Endian; kolejność bajtów w słowie: Big Endian (SwapBytes)</p> <p>2 - kolejność słów: Big Endian; kolejność bajtów w słowie: Little Endian (SwapWords)</p> <p>3 - kolejność słów: Little Endian; kolejność bajtów w słowie: Little Endian (NoSwap)</p>
BitFieldWidth	Liczba bitów pola bitowego. Suma <code>BitFieldWidth</code> i <code>BitFieldPosition</code> powinna być \leq <code>DataWidth</code> ; 0 oznacza brak pola bitowego (pełna szerokość danych = <code>DataWidth</code>)
BitFieldPosition	Pozycja najmłodszego bitu pola bitowego. Suma <code>BitFieldWidth</code> i <code>BitFieldPosition</code> powinna być \leq <code>DataWidth</code> .
Divisor	Dzielnik

Nazwa	Opis
InitialValueAccess	Metoda pierwszego dostępu do wartości Value: 0 - wartość początkowa Value jest odczytywana z urządzenia 1 - wartość początkowa Value jest zapisywana do urządzenia
InitialValue	Określa wartość początkową
Value	Zwraca ostatnio odczytaną wartość
RawValue	Nieprzeskalowana wartość rejestru
IsValid	Określa, czy wartość jest zgodna ze stanem obiektu
ErrorCode	Kod błędu: 1 - niedozwolona funkcja 2 - niedozwolony numer rejestru 3 - niedozwolona wartość danej 4 - uszkodzenie w przyłączonym urządzeniu 5 - potwierdzenie pozytywne 6 - brak gotowości, komunikat usunięty 7 - potwierdzenie negatywne 8 - błąd parzystości pamięci 10 - ścieżka bramy niedostępna 11 - brak odpowiedzi urządzenia docelowego 0 - poprawny odczyt/zapis rejestru -2 - przekroczenie czasu odpowiedzi -3 - błąd ramki (problem ze zdekodowaniem odpowiedzi) -4 - nieoczekiwany rozmiar odpowiedzi -5 - nieoczekiwany kod odpowiedzi -6 - nieprawidłowy stan obiektu -7 - błąd połączenia

METODY

Nazwa	Opis
<code>SetDeviceAddress</code>	Ustawia adres urządzenia Modbus
<code>SetResponseTimeout</code>	Ustawia czas przekroczenia odpowiedzi w krokach 25ms
<code>SetRefreshPeriod</code>	Ustawia okres odświeżania w krokach 5ms - wartość 0 wyłącza automatyczne odświeżanie
<code>SetRegisterAddress</code>	Ustawia adres obsługiwanego rejestru
<code>SetRegisterType</code>	Ustawia typ rejestru Modbus
<code>SetAlwaysWriteMultiple</code>	Zawsze używaj funkcji 15 lub 16 do zapisu wartości
<code>SetInputOutputCount</code>	Ustawia liczbę dyskretnych wejść / wyjść podlegających operacji odczytu / zapisu
<code>SetDataType</code>	Ustawia typ zmiennej
<code>SetDataWidth</code>	Ustawia szerokość danych
<code>SetEndianness</code>	Ustawia typ kolejności bajtów
<code>SetBitFieldWidth</code>	Ustawia liczbę bitów pola - 0 oznacza brak pola bitowego (pełna szerokość danych <code>DataWidth</code>)
<code>SetBitFieldPosition</code>	Ustawia pozycję najmłodszego bitu pola
<code>SetDivisor</code>	Ustawia dzielnik
<code>ReadValue</code>	Rozpoczyna odczyt wartości z urządzenia - czeka na zakończenie odczytu w przypadku braku <code>OnValueRead</code>
<code>WriteValue</code>	Zapisuje nową wartość do urządzenia

ZDARZENIA

Nazwa	Opis
<code>OnValueChange</code>	Zdarzenie wywołane w przypadku zmiany wartości cechy Value przez Modbus
<code>OnValueRead</code>	Zdarzenie wywołane po zakończeniu odczytu rozpoczętego przez metodę <code>ReadValue</code>
<code>OnValueWritten</code>	Zdarzenie wywołane po zakończeniu zapisu rozpoczętego przez metodę <code>WriteValue</code>
<code>OnError</code>	Zdarzenie wywołane gdy serwer zgłasza błąd

F. Obiekt ModbusServer

Uwaga!

Opisany obiekt wirtualny jest dostępny dla **GRENTON GATE MODBUS MASTER, DIN, Eth (INT-201 - E-01)** posiadający **firmware 1.4.1 - 2334 lub wyższy!**

CECHY

Nazwa	Opis
Port	Port nasłuchiwania serwera
DeviceAddress	Adres urządzenia Modbus
RegisterAddress	Adres obsługiwanego rejestru
RegisterType	Typ rejestru Modbus: 0 - wyjścia dwustanowe (coils) - funkcja Modbus: 5 (zapis pojedynczego wyjścia), 15 (zapis wielu wyjść) lub 1 (odczyt stanu wyjść) 1 - wejścia dwustanowe - funkcja Modbus: 2 2 - rejestry pamiętające - funkcja Modbus: 6 (zapis pojedynczego rejestru), 16 (zapis wielu rejestrów) lub 3 (odczyt rejestrów) 3 - rejestry wejściowe - funkcja Modbus: 4
DataType	Typ wartości: 0 - Liczba całkowita, stałoprzecinkowa lub pole bitowe bez bitu znaku 1 - Liczba całkowita, stałoprzecinkowa lub pole bitowe z bitem znaku 2 - Liczba zmiennoprzecinkowa
DataWidth	Szerokość danych (od 1 do 4 rejestrów 16 bitowych)
Endianness	Kolejność bajtów: 0 - kolejność słów: Big Endian; kolejność bajtów w słowie: Big Endian (SwapBytesAndWords) 1 - kolejność słów: Little Endian; kolejność bajtów w słowie: Big Endian (SwapBytes) 2 - kolejność słów: Big Endian; kolejność bajtów w słowie: Little Endian (SwapWords) 3 - kolejność słów: Little Endian; kolejność bajtów w słowie: Little Endian (NoSwap)
InitialValue	Określa wartość początkową
Value	Wartość rejestru
RawValue	Nieprzeskalowana wartość rejestru

METODY

Nazwa	Opis
SetValue	Ustawia wartość rejestru

ZDARZENIA

Nazwa	Opis
OnValueChanged	Zdarzenie wywoływane w przypadku zmiany wartości cechy <code>Value</code> przez Modbus

G. Obiekt Modbus

Uwaga!

Obiekt wirtualny `Modbus` został wycofany od wersji **firmware 1.4.1 - 2334**. W celu zachowania kompatybilności możliwe jest używanie obiektu jedynie w wcześniej utworzonych projektach.

CECHY

Nazwa	Opis
DeviceAddress	Adres urządzenia Modbus typu Slave
AccessRights	Tryb pracy: <i>read</i> (0 - odczyt); <i>read/write</i> (1 - odczyt/zapis)
RegisterAddress	Adres obsługiwanego rejestru
TransmissionSpeed	Prędkość transmisji
ValueType	Typ zmiennej (1 - <i>number</i> ; 2 - <i>float</i> ; 3 - <i>bit</i>)
BitPosition	Pozycja bitu (dotyczy dostępu bitowego do 16-bitowych rejestrów)
BitCount	Liczba bitów rejestru do odczytania
RefreshInterval	Czas odświeżania
ResponseTimeout	Czas na odpowiedź
Divisor	Dzielnik
Endianness	Kolejność bajtów i słów: <input type="checkbox"/> No swap (0) - bez zamiany; <input type="checkbox"/> Swap bytes and words (1) - zamiana kolejności bajtów i słów; <input type="checkbox"/> Swap bytes (2) - zamiana kolejności bajtów w obrębie każdego ze słów; <input type="checkbox"/> Swap words (3) - zamiana słów)
RegisterType	Typ rejestru Modbus: <input type="checkbox"/> 0 - wejścia/wyjścia bitowe, <input type="checkbox"/> 1 - wejścia dwustanowe, <input type="checkbox"/> 2 - rejestry pamiętające, <input type="checkbox"/> 3 - rejestry wejściowe
ErrorCode	Kod błędu: <input type="checkbox"/> -3 - błąd ramki; <input type="checkbox"/> -2 - przekroczenie czasu odpowiedzi; <input type="checkbox"/> -1 - nieaktualna wartość ostatniego odczytanego rejestru; <input type="checkbox"/> 0 - poprawny odczyt/zapis rejestru; <input type="checkbox"/> 1 - niedozwolona funkcja; <input type="checkbox"/> 2 - niedozwolony numer rejestru; <input type="checkbox"/> 3 - niedozwolona wartość danej; <input type="checkbox"/> 4 - uszkodzenie w przyłączonym urządzeniu; <input type="checkbox"/> 5 - potwierdzenie pozytywne; <input type="checkbox"/> 6 - brak gotowości/komunikat usunięty; <input type="checkbox"/> 7 - potwierdzenie negatywne; <input type="checkbox"/> 8 - błąd parzystości pamięci
Value	Wartość odczytu / zapisu
RegisterValue	Nieprzeskalowana wartość rejestru

Nazwa	Opis
StopBits	Bity stopu: 0 - 1 bit stopu 1 - 1.5 bitu stopu 2 - 2 bity stopu
Parity	Bit parzystości: 0 - None 1 - Odd 2 - Even

METODY

Nazwa	Opis
SetDeviceAddress	Ustawia adres urządzenia Modbus typu Slave
SetAccessRights	Ustawia tryb pracy: odczyt lub odczyt/zapis
SetRegisterAddress	Ustawia adres obsługiwanego rejestru
SetTransmissionSpeed	Ustawia prędkość transmisji
SetValueType	Ustawia typ zmiennej
SetBitPosition	Ustawia pozycję bitu
SetBitCount	Ustawia liczbę bitów rejestru do odczytania
SetRefreshInterval	Ustawia czas odświeżania
SetResponseTimeout	Ustawia czas oczekiwania na odpowiedź
SetDivisor	Ustawia dzielnik
SetEndianness	Ustawia typ kolejności bajtów
SetRegisterType	Ustawia typ rejestru Modbus
SetValue	Ustawia wartość odczytu / zapisu
SetStopBits	Ustawia liczbę bitów stopu
SetParity	Ustawia sprawdzanie parzystości

ZDARZENIA

Nazwa	Opis
OnChange	Zdarzenie wywoływane przy zmianie stanu (niezależnie od wartości)
OnError	Zdarzenie wywołane, gdy urządzenie slave zgłasza błąd

H. Obiekt ModbusValue

Uwaga!

Obiekt wirtualny `ModbusValue` został wycofany od wersji **firmware 1.4.1 - 2334**. W celu zachowania kompatybilności możliwe jest używanie obiektu jedynie w wcześniej utworzonych projektach.

CECHY

Nazwa	Opis
TransmissionSpeed	Prędkość transmisji
Parity	Bit parzystości: <input type="radio"/> 0 - None <input type="radio"/> 1 - Odd <input type="radio"/> 2 - Even
StopBits	Bity stopu: <input type="radio"/> 0 - 1 bit stopu <input type="radio"/> 1 - 1.5 bitu stopu <input type="radio"/> 2 - 2 bity stopu
DeviceAddress	Adres urządzenia Slave Modbus
ResponseTimeout	Czas na odpowiedź w krokach 25ms
RefreshPeriod	Minimalny okres odświeżania w krokach 5ms. Wartość 0 wyłącza automatyczne odświeżanie
RegisterAddress	Adres obsługiwanego rejestru
RegisterType	Typ rejestru Modbus (0 - wejścia/wyjścia bitowe, 1 - wejścia dwustanowe, 2 - rejestry pamiętające, 3 - rejestry wejściowe)
InputOutputCount	Określa liczbę dyskretnych wejść / wyjść podlegających operacji odczytu / zapisu
DataType	Typ wartości: <input type="radio"/> 0 - Liczba całkowita, stałoprzecinkowa lub pole bitowe bez bitu znaku <input type="radio"/> 1 - Liczba całkowita, stałoprzecinkowa lub pole bitowe z bitem znaku <input type="radio"/> 2 - Liczba zmiennoprzecinkowa
DataWidth	Szerokość danych (od 1 do 4 rejestrów 16 bitowych)
Endianness	Kolejność bajtów: <input type="radio"/> 0 - kolejność słów: Big Endian; kolejność bajtów w słowie: Big Endian <input type="radio"/> 1 - kolejność słów: Little Endian; kolejność bajtów w słowie: Big Endian <input type="radio"/> 2 - kolejność słów: Big Endian; kolejność bajtów w słowie: Little Endian <input type="radio"/> 3 - kolejność słów: Little Endian; kolejność bajtów w słowie: Little Endian
BitFieldWidth	Liczba bitów pola bitowego. Suma <input type="text"/> i <input type="text"/> powinna być \leq <input type="text"/> ; 0 oznacza brak pola bitowego (pełna szerokość danych = <input type="text"/>)
BitFieldPosition	Pozycja najmłodszego bitu pola bitowego. Suma <input type="text"/> i <input type="text"/> powinna być \leq <input type="text"/> .

Nazwa	Opis
Divisor	Dzielnik (skala) wartości
InitialValueAccess	Metoda pierwszego dostępu do wartości Value: 0 - wartość początkowa Value jest odczytywana z urządzenia 1 - wartość początkowa Value jest zapisywana do urządzenia
Value	Zwraca ostatnio odczytaną wartość oraz określa wartość początkową
RawValue	Nieprzeskalowana wartość rejestru
IsValid	Określa, czy wartość jest zgodna ze stanem obiektu
ErrorCode	Kod błędu: 1 - niedozwolona funkcja 2 - niedozwolony numer rejestru 3 - niedozwolona wartość danej 4 - uszkodzenie w przyłączonym urządzeniu 5 - potwierdzenie pozytywne 6 - brak gotowości, komunikat usunięty 7 - potwierdzenie negatywne 8 - błąd parzystości pamięci 0 - poprawny odczyt/zapis rejestru -2 - przekroczenie czasu odpowiedzi -3 - błąd ramki (problem ze zdekodowaniem odpowiedzi) -4 - nieoczekiwany rozmiar odpowiedzi -5 - nieoczekiwany kod odpowiedzi

METODY

Nazwa	Opis
SetTransmissionSpeed	Ustawia prędkość transmisji
SetParity	Ustawia sprawdzanie parzystości
SetStopBits	Ustawia liczbę bitów stopu
SetDeviceAddress	Ustawia adres urządzenia Modbus typu Slave
SetResponseTimeout	Ustawia czas przekroczenia odpowiedzi w krokach 25ms
SetRefreshPeriod	Ustawia okres odświeżania w krokach 5ms - wartość 0 wyłącza automatyczne odświeżanie
SetRegisterAddress	Ustawia adres obsługiwanego rejestru
SetRegisterType	Ustawia typ rejestru Modbus
SetInputOutputCount	Ustawia liczbę dyskretnych wejść / wyjść podlegających operacji odczytu / zapisu
SetDataType	Ustawia typ zmiennej
SetDataWidth	Ustawia szerokość danych
SetEndianness	Ustawia typ kolejności bajtów
SetBitFieldWidth	Ustawia liczbę bitów pola - 0 oznacza brak pola bitowego (pełna szerokość danych <code>DataWidth</code>)
SetBitFieldPosition	Ustawia pozycję najmłodszego bitu pola
SetDivisor	Ustawia dzielnik
ReadValue	Rozpoczyna odczyt wartości z urządzenia - czeka na zakończenie odczytu w przypadku braku OnValueRead
WriteValue	Zapisuje nową wartość do urządzenia

ZDARZENIA

Nazwa	Opis
OnValueChange	Zdarzenie wywoływane w przypadku zmiany wartości cechy Value lub zmiany parametrów obiektu
OnValueRead	Zdarzenie wywoływane po zakończeniu odczytu rozpoczętego przez metodę ReadValue
OnError	Zdarzenie wywoływane gdy urządzenie slave zgłasza błąd