RS485 DO ETH (B)
Podręcznik MQTT i JSON

ZAWARTOŚĆ

1. PRZEGLĄD

MQTT i JSON mogą być używane samodzielnie lub razem. Wśród nich JSON obsługuje Modbus przekonwertuj format RTU na format JSON.

Główne cechy:

1. Aby nawiązać połączenie z serwerem, użyj protokołu opartego na MQTT i skorzystaj z formularza subskrypcji na publikację transmisji danych.
2. Wsparcie niezależnego projektowania i automatycznego gromadzenia rejestrów Modbus RTU.
3. Obsługuj konwersję określonej zawartości rejestru Modbus do formatu JSON i wyślij ją regularnie i aktywnie.
4. Obsługa dodawania identyfikatora urządzenia, godziny i dowolnego ciągu w formacie JSON.
5. Obsługa zagnieżdżonej metody zapisu w formacie JSON.
6. Obsługuj protokół NTP, uzyskaj czas automatycznie.
7. Obsługuje dane bez znaku i dane ze znakiem, obsługuje reprezentację przecinka dziesiętnego i obsługuje dane o długości 4 bajtów.
8. Wszystkie konfiguracje można przeprowadzić w konfiguracji interfejsu i są one niezależne od użytkownika konfiguracja nie wymaga dostosowywania.
9. Oprócz wyboru MQTT, protokół może obsługiwać metody HTTP POST i GET.

2. JSON PROSTY PRZYKŁAD

2.1. MODBUS RTU DO JSON

ZLAN Modbus RTU do JSON może realizować automatyczne zbieranie tabel Modbus RTU i podążać

Format JSON jest automatycznie przesyłany na serwer w chmurze.

Tutaj wyjaśniamy to użycie na konkretnym przypadku.

2.2. TABELA MODBUSA

Załóżmy, że istnieje tabela Modbus z kodem funkcji 3 i adresem 1. Jest adresy rejestrów i nazwy parametrów są następujące. Oznacza to, że długość bajtu wynosi 4 że 2 rejestry muszą być odczytywane w sposób ciągły.

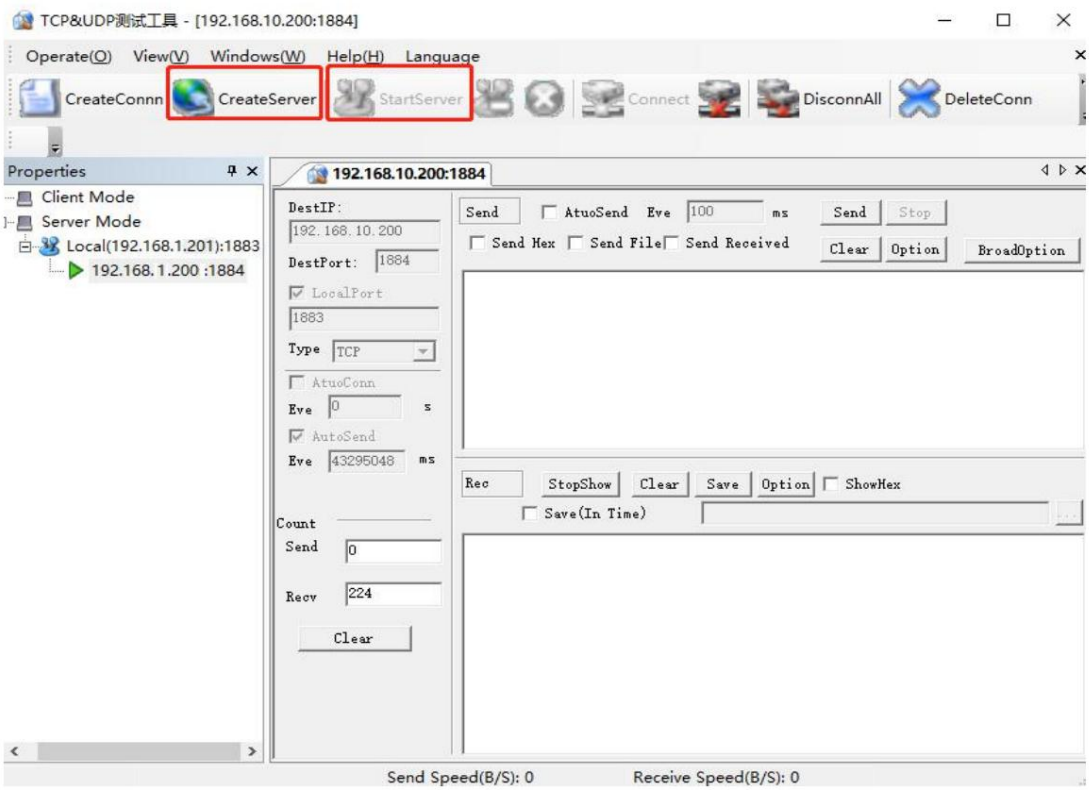
Rejestr adres	Nazwa parametru	Bajt długość	Uwagi
0	Bieżąca suma aktywnych energia	4	Bez znaku, zachowaj 2 miejsca po przecinku miejsca
97	Napięcie fazy A	2	Bez znaku, 1 miejsce po przecinku skryty
98	Napięcie fazy B	2	
99	Napięcie fazy C	2	
100	Prąd fazy A	2	Bez znaku, 2 miejsca po przecinku skryty
101	Prąd fazy B	2	
102	Prąd fazy C	2	
119	Częstotliwość	2	
356	Fazowa moc czynna	4	Bez znaku, 3 miejsca po przecinku skryty
358	Moc czynna fazy B	4	
360	Moc czynna fazy C	4	
362	Całkowita moc czynna	4	

Tak zwany znak ze znakiem oznacza, że najwyższy bit 2 bajtów lub 4 bajtów jest bitem znaku, np na przykład 0xFFFF zostanie rozpoznane jako -1. Zachowanie 2 miejsc po przecinku oznacza, że po danych jest konwertowany na liczbę całkowitą, przecinek dziesiętny przesuwa się od prawej do lewej 2 cyfr.

2.3. KONFIGURACJA URZADZENIA

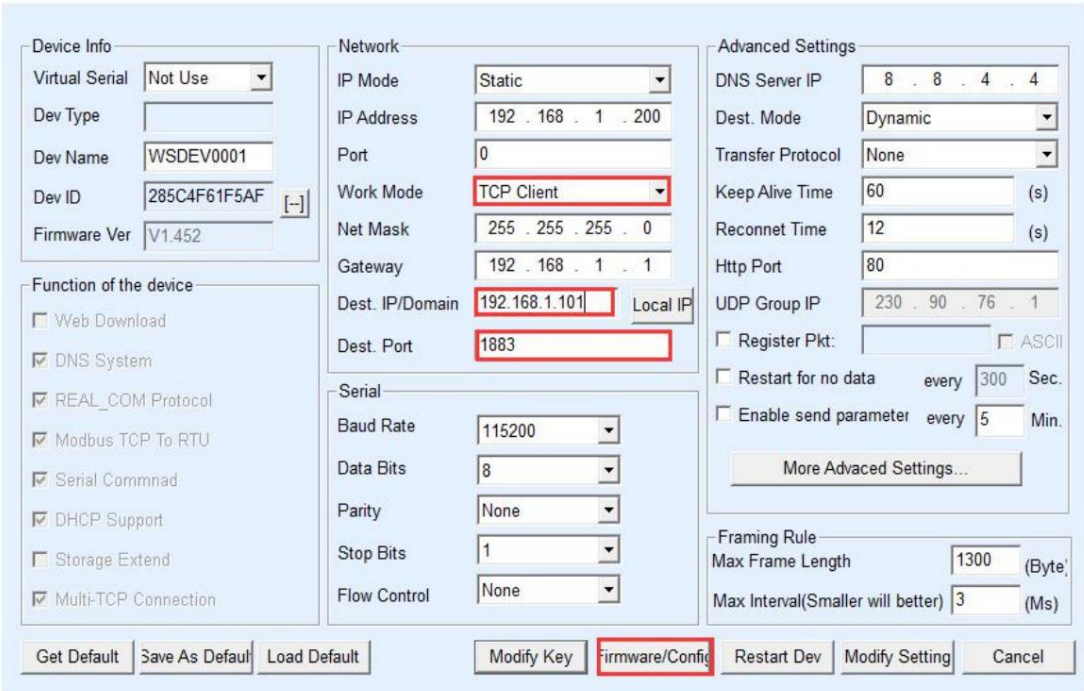
Konfigurujemy urządzenie jako klient.

Użyj narzędzia portu szeregowego, aby monitorować serwer TCP na porcie 1883 komputera lokalnego.



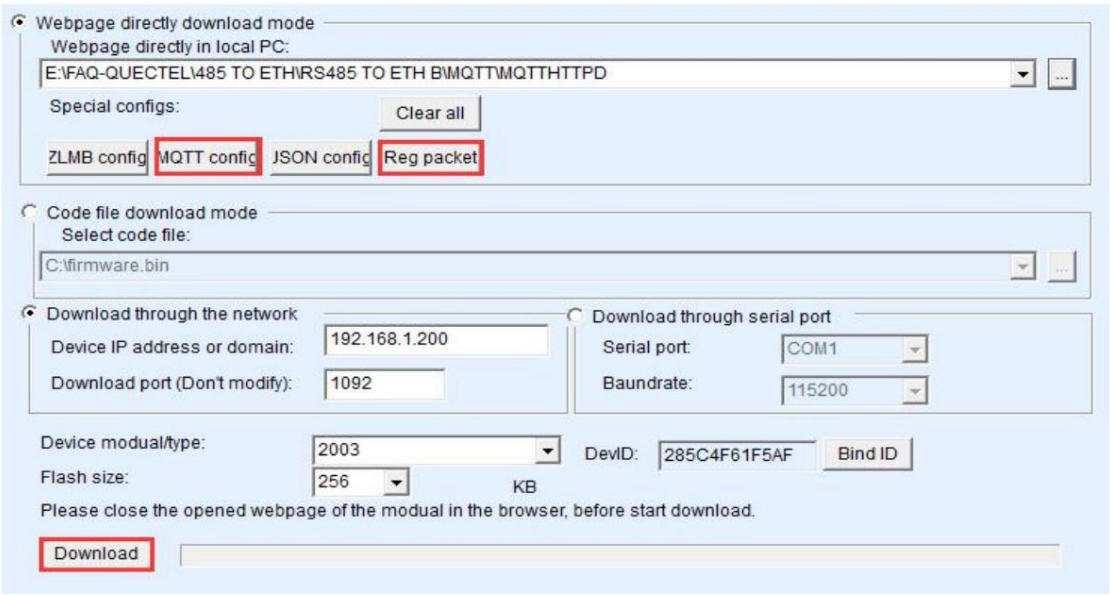
Rysunek 1 Serwer symulowany gniazdem odbierający dane

Do konfiguracji urządzenia użyj ZLVircom.



Rysunek 2 Konfiguracja urządzenia

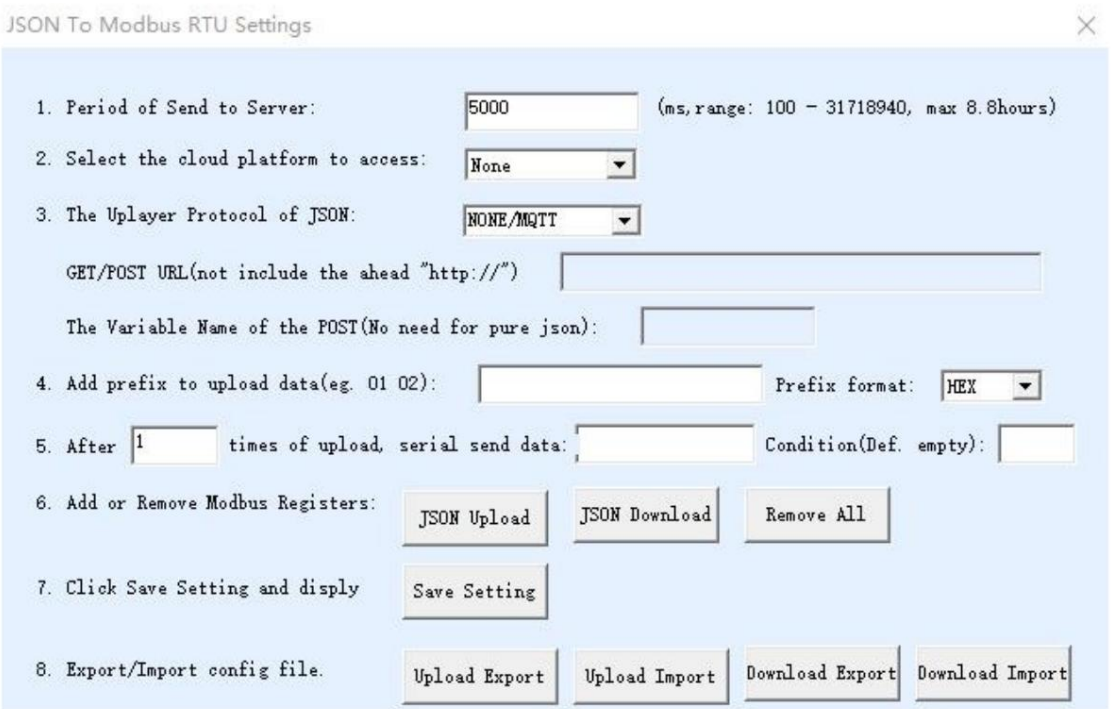
Kliknij Modyfikuj konfigurację, aby podłączyć urządzenie do narzędzia SocketDlgTest. Wprowadź urządzenie ponowne edytowanie okna dialogowego. Kliknij przycisk „Oprogramowanie sprzętowe i konfiguracja”.



Rysunek 3 Interfejs pobierania

Najpierw kliknij „Pobieranie katalogu internetowego”, aby przejść do trybu pobierania konfiguracji. Następnie wybierz nowy pusty katalog, taki jak katalog MQTTHTPD. Aby zapobiec poprzedniemu projekt z pozostałych, kliknij najpierw przycisk „Wyczyść wszystko”, aby wyświetlić poprzedni projekt zawartość można wyczyścić. Plik projektu zostanie zapisany w tym katalogu i będzie można go pobrać na urządzenie, klikając później przycisk „pobierz”.

Kliknij przycisk „Konfiguracja JSON”.



Rysunek 4 Główny interfejs konfiguracyjny JSON

Parametry są tutaj następujące:

1. Czas przesyłania serwera: Za każdym razem do serwera wysyłane są domyślne dane JSON. Serwer jest docelowy adres IP ustawiony właśnie w interfejsie konfiguracyjnym urządzenia, a urządzenie jest milisekundy.

2. Dodaj/Wyświetl: Po kliknięciu możesz zaprojektować każdy węzeł JSON lub wyświetlić aktualnie zaprojektowane treści.

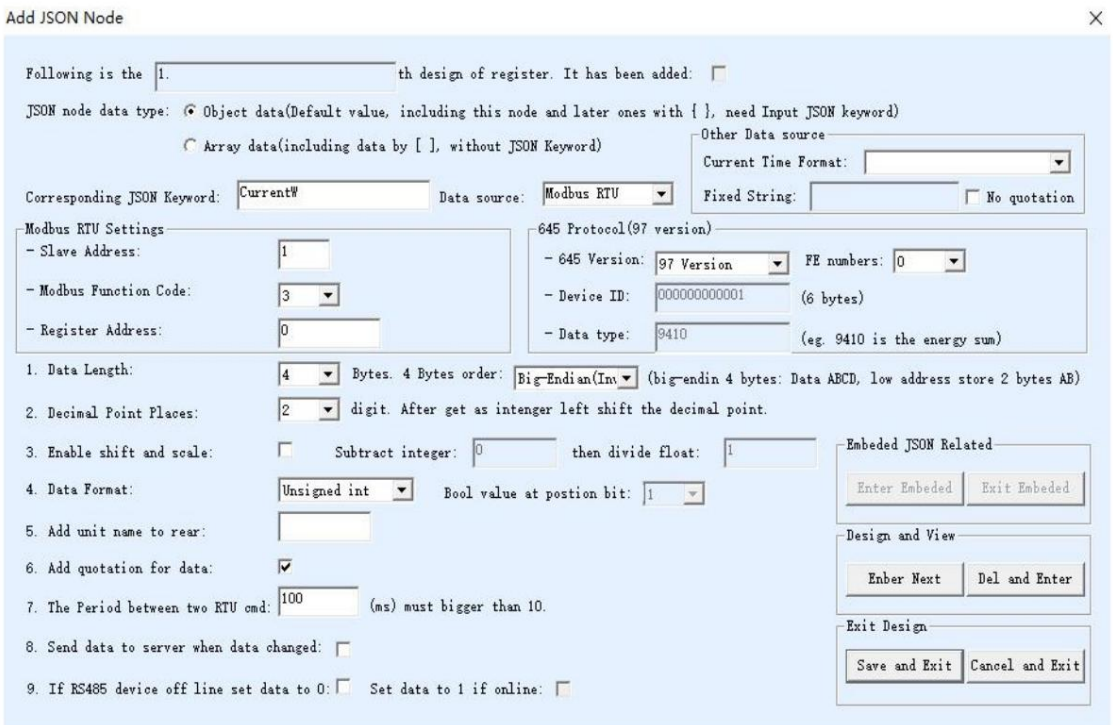
3. Usuń wszystko: Usuń wszystkie rejestry Modbus zaprojektowane za pomocą przycisku „Dodaj/Wyświetl”, aby ułatwić ponowne uruchomienie projektu.

4. Zapisz ustawienia JSON: Po zakończeniu projektowania wystarczy kliknąć ten przycisk, aby zapisać dane do katalogu pobierania, a następnie pobierz go do wnętrza urządzenia.

Teraz kliknij przycisk „Dodaj/Wyświetl”. Dla pierwszego wiersza poprzedniej tabeli Modbus:

Rejestr adres	Nazwa parametru	Bajt długość	Uwagi
0	Bieżąca suma aktywnych energia	4	Bez znaku, zachowaj 2 miejsca po przecinku miejsca

Odpowiednia konfiguracja jest następująca:



Rysunek 5 Ustawienia rejestru

Parametry są tutaj następujące:

1. Poniższy rysunek przedstawia pierwsze słowo kluczowe JSON: „1”. tutaj oznacza kilka pierwszych JSON słowa kluczowe bieżącego interfejsu projektu, jeśli drugie to „2.”, jeśli jest to węzeł zagnieżdżony w drugim JSON To jest „2.1” i tak dalej.

2. Już dodano: Jeśli jest zaznaczone, oznacza to, że zostało dodane. Podczas przeglądania skonfigurowanych informacji, pojawi się znacznik wskazujący, że jest on w stanie edycji. Jeśli istnieje niezaznaczony, jest w stanie dodanym.

3. Odpowiednie słowo kluczowe JSON: nazwa tego węzła JSON.

4. Źródło danych: wybierz źródło danych JSON

a) Modbus RTU: Na przykład w postaci CurrentW: 123,45 oznacza to, że dane pochodzi z określonej tabeli Modbus RTU i jest zbierany przez port szeregowy. Lewo połowa rysunku związana jest z projektowaniem parametrów Modbus RTU.

b) Stały ciąg znaków: Na przykład w formie DevName: „MyDev” wpisz MyDev w stałym string po prawej stronie, a nazwa JSON to DevName, więc będzie to stały ciąg węzłów JSON można wygenerować.

c) Identyfikator urządzenia: Jeśli nazwa węzła JSON to DevID, ciąg wysłanego węzła to DevID: „285301020304”, gdzie „285301020304” to adres MAC lub unikalny numer urządzenie.

d) Bieżący czas: Jeśli nazwa węzła JSON to ColletTime, przesłany ciąg znaków to ColletTime: „2019-05-13 22:23:31”. Czas to czas uzyskany przez system poprzez NTP protokół.

e) Osadzanie JSON: Jeśli nazwa węzła to Alarm, format jego przesyłania ma

Alarm: {temp1: "25.1", temp2: "26.2"}, czyli zawartość Alarmu nadal jest w formacie JSON kolekcja

5. Ustawienia związane z Modbus

a) Adres urządzenia podrzędnego: Adres tabeli Modbus.

b) Kod funkcji Modbus: obecnie obsługuje kody funkcji 03 i 04.

c) Adres rejestru: tutaj odpowiednie 0.

d) Długość danych: odpowiada 4 bajtom.

e) Format danych: Odpowiada liczbie całkowitej bez znaku.

f) Zachowaj przecinek dziesiętny: zarezerwowane są tutaj 2 cyfry.

g) Zwiększ jednostkę po danych: Na przykład, gdy „Prąd W”: 25,6 W, W za
25,6 to duża jednostka, którą należy zwiększyć. Wpisz „W” w tym polu.

h) Dodaj dane w cudzysłowie: jeśli zaznaczone, zmień „CurrentW”: 25,6W na „CurrentW”:
Forma „25,6 W”.

i) Czas odpytywania portu szeregowego: tutaj ustawiony na 100 ms. Odnosi się do odpytywania tego rejestru i
następny rejestr zamiast interwału odpytywania dla tego polecenia.

6. Stały ciąg: Gdy źródło jest wybrane jako stały ciąg, możesz wprowadzić treść
sznurek.

7. Przycisk

a) Zagnieżdżony JSON: Gdy bieżące źródło węzła zostanie wybrane jako typ „zagnieżdżony JSON”, ty
należy kliknąć ten przycisk, aby wejść do projektu zagnieżdżonego JSON, jeśli obecnie jest to „2.”, wejdzie
projekt węzła „2.1” .

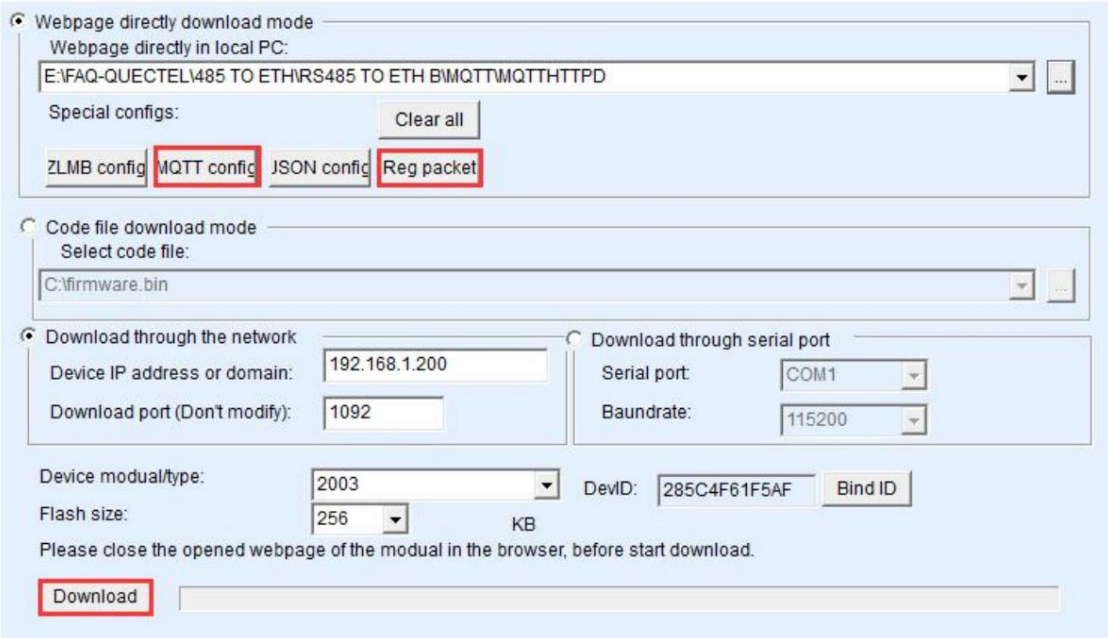
b) Wróć do poprzedniego poziomu: Jeśli bieżący węzeł jest zagnieżdżony na N-tym poziomie, kliknij to
przycisk powróci do projektu węzła poziomu N-1 i pozostanie na nowym węźle na poziomie N-1
poziom.

c) Zaprojektuj następny: Kliknij, aby wejść do kolejnego lokalnego węzła JSON. Jeśli nie ma następnego węzła w
poprzedniego projektu, pole wyboru „Już dodano” zostanie usunięte, co oznacza, że jest to a
nowy węzeł.

d) Zapisz projekt: Po ukończeniu projektu kliknij „Zapisz projekt” w ostatnim węźle projektowania
interfejs. Następnie wróć do głównego interfejsu i kliknij „Zapisz konfigurację JSON”.

e) Anuluj projekt: anuluj wszystkie bieżące projekty, jeśli przeglądasz treść projektu, jest to możliwe
kliknij ten przycisk, aby wyjść

Tutaj kliknij przycisk „Zaprojektuj dalej”, aby kontynuować projektowanie innych rejestrów w
Tabela Modbusa. Po zaprojektowaniu wszystkich rejestrów w formularzu kliknij „Zakończ projekt”, a następnie
kliknij „Zapisz konfigurację JSON”, aby wyjść. Następnie kliknij przycisk „Pobierz” w pliku „Pobierz
Strona internetowa

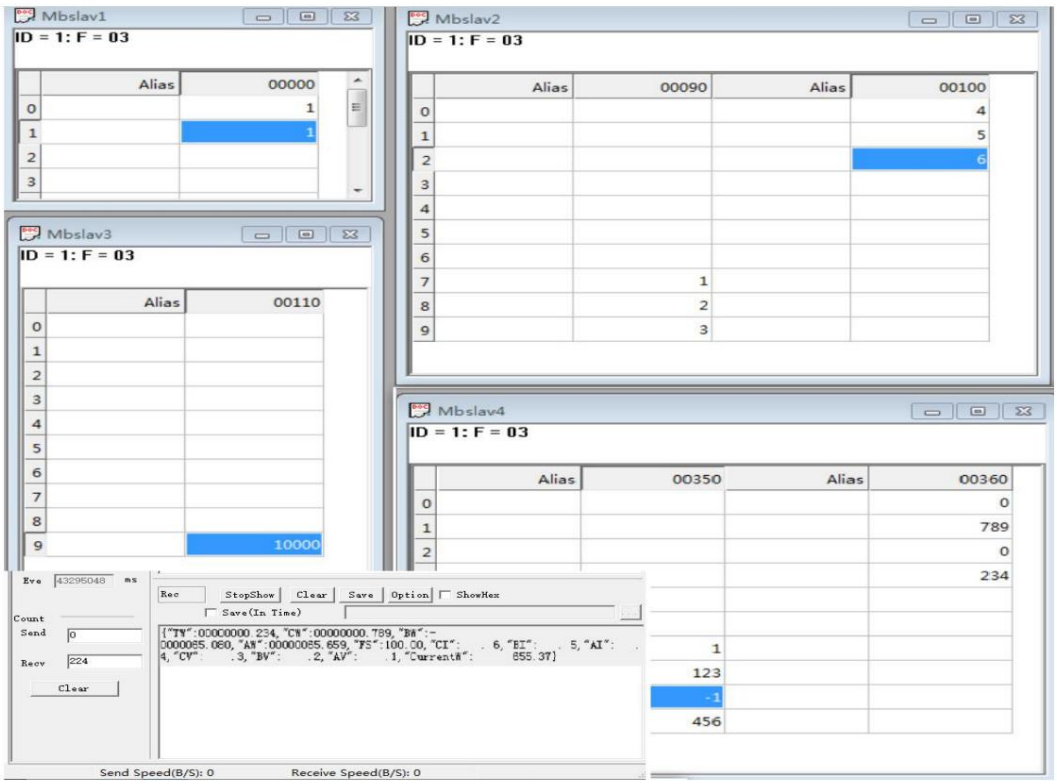


Rysunek 6 Pobieranie

Następnie kliknij „OK”, a urządzenie automatycznie uruchomi się ponownie. Jeśli nie ma ponownego uruchomienia, proszę uruchom ponownie ręcznie.

2.4. STWÓRZ NOWY MIERNIK ANALOGOWY MODBUS

Tutaj Modbus Slave służy do symulacji licznika



Rysunek 7 Wyniki testu

Wynik testu pokazuje, że licznik może być symulowany za pomocą narzędzia Modbus Slave zbierane przez bramkę. jednocześnie można go wysłać do symulowanego oprogramowania serwera regularnie przez SocketDlgTest zgodnie z formatem json.

3. Przykład złożony JSON

3.1. WŁĄCZ NTP

Aby móc używać JSON z czasem, musisz najpierw włączyć funkcję NTP pliku urządzenie. Funkcja NTP może uzyskać aktualny czas przez sieć.

W katalogu pobierania internetowego, czyli katalogu, w którym znajduje się plik httpd.txt, utwórz plik pusty plik txt z zawartością w dół:

[NTP]

NTP_SERVER1=a1.a2.a3.a4

NTP_SERVER2=b1.b2.b3.b4

NTP_SERWER3=c1.c2.c3.c4

RE_ARUIRE_TIME=0

NTP_SERVER1, NTP_SERVER2, NTP_SERVER3 to adresy IP serwera czasu NTP Adres, wypełnić zgodnie ze stanem faktycznym. Można skonfigurować maksymalnie 3 serwery, ale jest to konieczne NTP_SERVER1 zaczyna pisać, jeśli jest tylko jeden, napisz NTP_SERVER1, jeśli są tylko dwa, pisać NTP_SERVER1 i NTP_SERVER2.

Po zapisaniu plik ntp.txt i inne pliki zostaną pobrane do wnętrza urządzenia.

3.2. ZAgnieżdżony projekt JSON

Załóżmy, że musimy zaprojektować następujący kod JSON:

```
{"nagłówek":{"DEVID":"285301020304",
                "czas":" 2019-05-13 22:23:31"},
  "dane": {"id": "MojeDane123456",
            "alarm":{"alarm1":123,4C
                    „alarm2”: 567,8C
                }
            },
```

„Wartość”: 2345

}

Etapy projektowania są następujące:

1. Słowem kluczowym w kroku 1 jest „nagłówek”, następnie wybierz „JSON Nesting” jako źródło, a następnie kliknij przycisk „Zaprojektuj zagnieżdżony JSON”.

2. Przejdź do kroku 1.1, tutaj zaprojektuj „DEVID”: „285301020304”, wprowadź słowo kluczowe jako DEVID, wybierz „Identyfikator urządzenia” jako źródło i kliknij „Zaprojektuj dalej”.

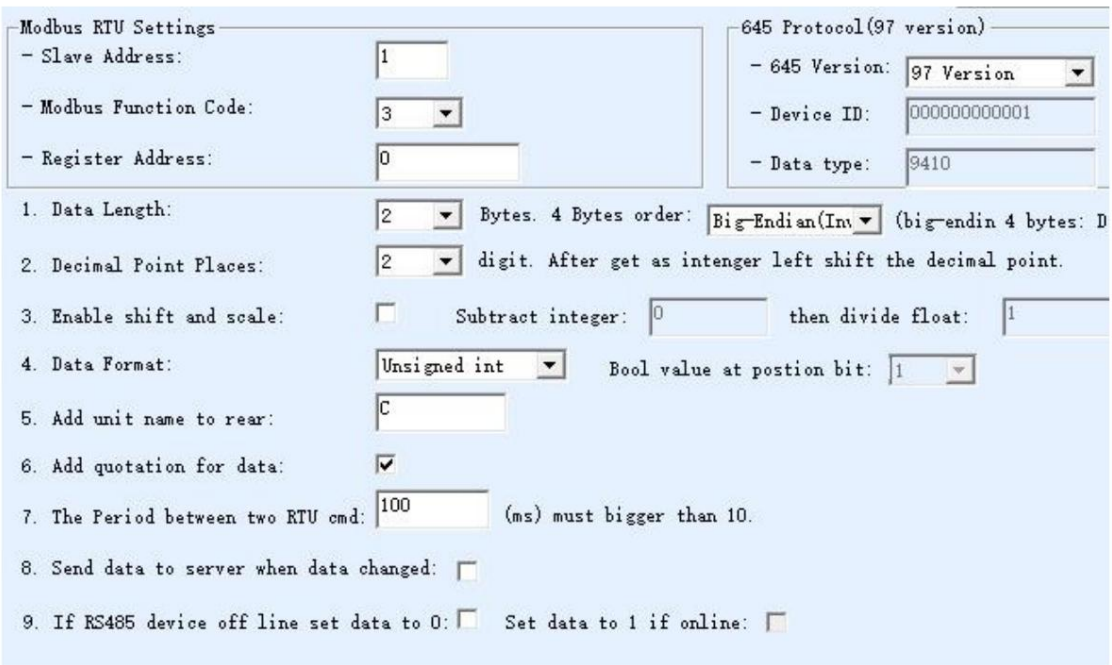
3. Wejdź do kroku 1.2, tutaj zaprojektuj „czas”: „2019-05-13 22:23:31”, wprowadź słowo kluczowe jako czas, wybierz „bieżący czas” jako źródło, należy zauważyć , że chociaż pierwszy poziom ma został zaprojektowany jako gotowy, ale nadal musisz kliknąć „Zaprojektuj dalej”. Po wejściu do kroku 1.3 kliknij „Wróć do poprzedniego poziomu”. Ten krok 1.3 zostanie automatycznie porzucony.

4. Przejdź do kroku 2. Wprowadź tutaj dane słowa kluczowego, następnie jako źródło wybierz „JSON Nesting” i następnie kliknij przycisk „Zaprojektuj zagnieżdżony JSON”.

5. Przejdź do kroku 2.1, tutaj zaprojektuj „id”: „MyData123456”, wprowadź identyfikator słowa kluczowego, wybierz źródło jako stały ciąg znaków, następnie wpisz „MyData123456” w polu stałego ciągu i kliknij „Zaprojektuj Następny".

6. Przejdź do kroku 2.2, wprowadź tutaj słowo kluczowe alarm, jako źródło wybierz „JSON Nesting” i następnie kliknij przycisk „Zaprojektuj zagnieżdżony JSON”.

7. Przejdź do kroku 2.2.1, aby zaprojektować „alarm 1”: 123.4C, są to dane Modbus z jednostką, funkcja kod to 3, rejestr to 0, wówczas projekt pokazano na rysunku:



Rysunek 8 Projekt rejestru

Następnie kliknij „Zaprojektuj dalej”.

8. Przejdź do kroku 2.2.2, aby zaprojektować „alarm2”: Metoda jest podobna do alarmu1, a rejestr adres jest ustawiony na 1. Podobnie, najpierw kliknij „Zaprojektuj dalej”, a następnie w kroku „2.2.3” kliknij „Powrót do poprzedniego poziomu”.

9. Wejdź do sekcji 2.3, ponieważ 2.3 nie trzeba projektować, kliknij „wróć do poprzedniego poziom” w tej chwili. Ponieważ nadal istnieje „wartość”: 2345 nie jest zaprojektowany, w przeciwnym razie możesz bezpośrednio „zapisz projekt”.

10. Przejdź do sekcji 3. Tutaj zaprojektuj dane Modbus ze słowem kluczowym. Teraz kliknij „Zapisz”. Projekt”. Uwaga: Jeśli „wartość”: 2345 nie istnieje tutaj, musisz bezpośrednio kliknąć Zapisz w poprzednim kroku. Jeśli przypadkowo kliknąłeś „powrót do poprzedniego poziomu”, aby wejść trzeci tutaj, ale trzeci nie istnieje, możesz z niego korzystać w tej chwili Nowa wersja zlvircom „Usuń i przejdź do następnego”, aby usunąć ten bezużyteczny węzeł.

11. Wróć do interfejsu ustawień JSON do Modbus RTU i kliknij „Zapisz ustawienia JSON”. Następnie pobierz go do wnętrza urządzenia i użyj go.

Po nawiązaniu połączenia TCP dane są odbierane:

{"header":{"DEVID":"2850002F0EEC","time":"2019-05-13

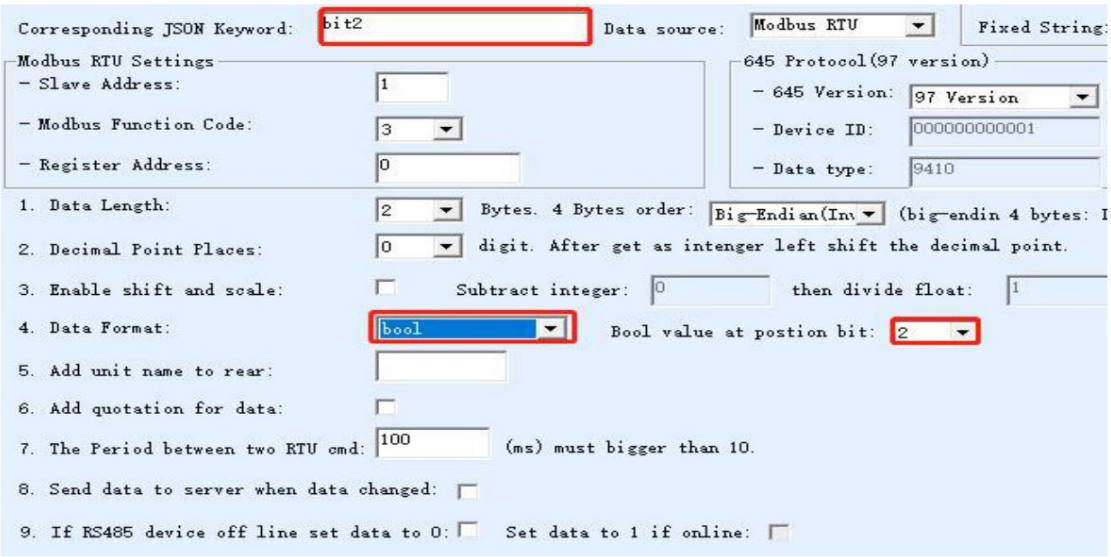
23:41:26"},"data":{"id":"MojeDane123456","alarm":{"alarm1": 123,4°C,"alarm2":"

567,8C"}},"wartość": 2345}

Pamiętaj, że jeśli edytujesz bieżący projekt JSON, musisz wybrać poprawny miejsce docelowe w interfejsie pobierania internetowego. Ponadto kliknij odpowiedni przycisk z etapami projektowania, aby w pełni przeglądać wszystkie węzły.

3.3. PRZECZYTAJ BITY REJESTRU BAJTOWEGO

Czasami dane odczytane przez Modbus przy użyciu kodu funkcji 03/04 również będą wymagały bitów wyrażać określone znaczenia. Na przykład rejestr adresu 00 00 odczytany przez 03 kod funkcji to 0x8183, następnie uwzględniane są bit16, bit9, bit8, bit2 i bit1. Wszystkie są 1. Te bity mają różne znaczenia. Gdy wynoszą 1, wskazują różne alarmy. Więc ty także należy przesłać z różnymi słowami kluczowymi JSON. Ta funkcja wymaga oprogramowania sprzętowego 2003 1.579 i nowszych i jest zaprojektowany z zlvircom 5.13 i nowszymi. Metody jak poniżej:

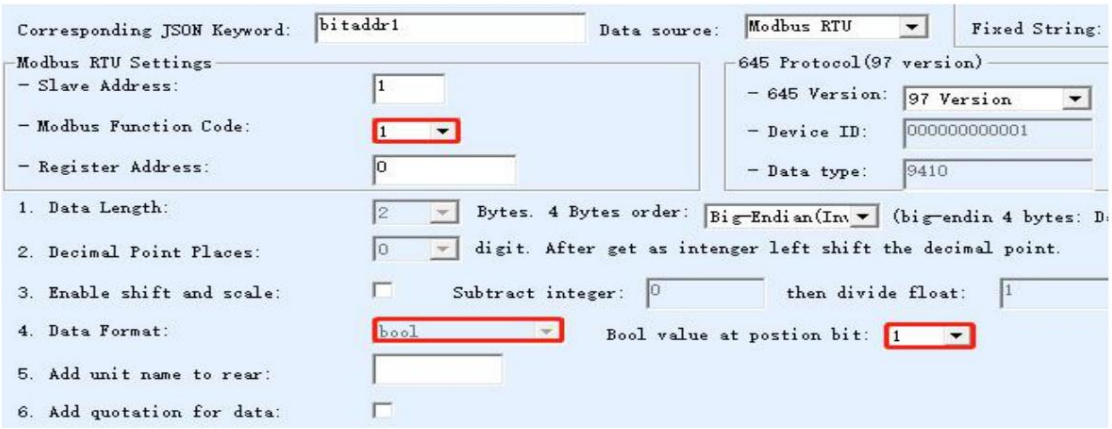


Rysunek 9 Rejestr bajtowy

Metoda projektowania jest w zasadzie taka sama jak poprzednia metoda. Jedyne co trzeba zapłacić
Należy zwrócić uwagę, że format danych jest wybrany jako „Boolean”, a następnie tam, gdzie jest to wartość Boolean
wartość jest zlokalizowana, bit2 jest zmienną json w kodzie funkcji 03 i pozycją bitu w rejestrze 00. Jeśli wartość rejestru
wynosi 00 20, gdzie 1 znajduje się na pozycji 2, należy pamiętać, że można ustawić ten sam adres stacji Modbus, kod
funkcji i zarejestrować się dla różnych słów kluczowych json.
Można uzyskać różną zawartość zmiennej, o ile pozycja wartości logicznej jest inna. Na przykład zaprojektowaliśmy
bit1, bit2, bit8, bit9, bit16, gdy zawartość rejestru jest
0x8183, otrzymujemy następujący zwrot danych json: {"bit1":1,"bit2":1,"bit8":1,"bit9" :1,"bit16":1}

3.4. KODY FUNKCJI 01 I 02

Można ustawić węzeł JSON dla rejestru bitowego kodu funkcji 01/02, ale każdy
Węzeł JSON musi raz ustawić adres rejestru, więc liczba odczytywanych za każdym razem bitów wynosi
jeden kawałek. Różnica w stosunku do bitu rejestru bajtowego jest następująca: bit rejestru bajtowego jest nieruchomy
odczytaj kod funkcji 03/04, ale zostanie pobrana tylko wartość jednego z 2 bajtów; odczytywany jest sam kod funkcji
01/02. W przypadku wyboru bitu, ponieważ jednocześnie odczytywany jest tylko 1 bit, pozycja wartości logicznej
jest zwykle zapisywana z liczbą 1. Jeśli wynosi 1, wyświetli się „1”, w przeciwnym razie zostanie wyświetlić „0”.



Rysunek 10 Rejestr bajtowy

Gdy kod funkcji Modbus to 01/02, długość danych, format danych i

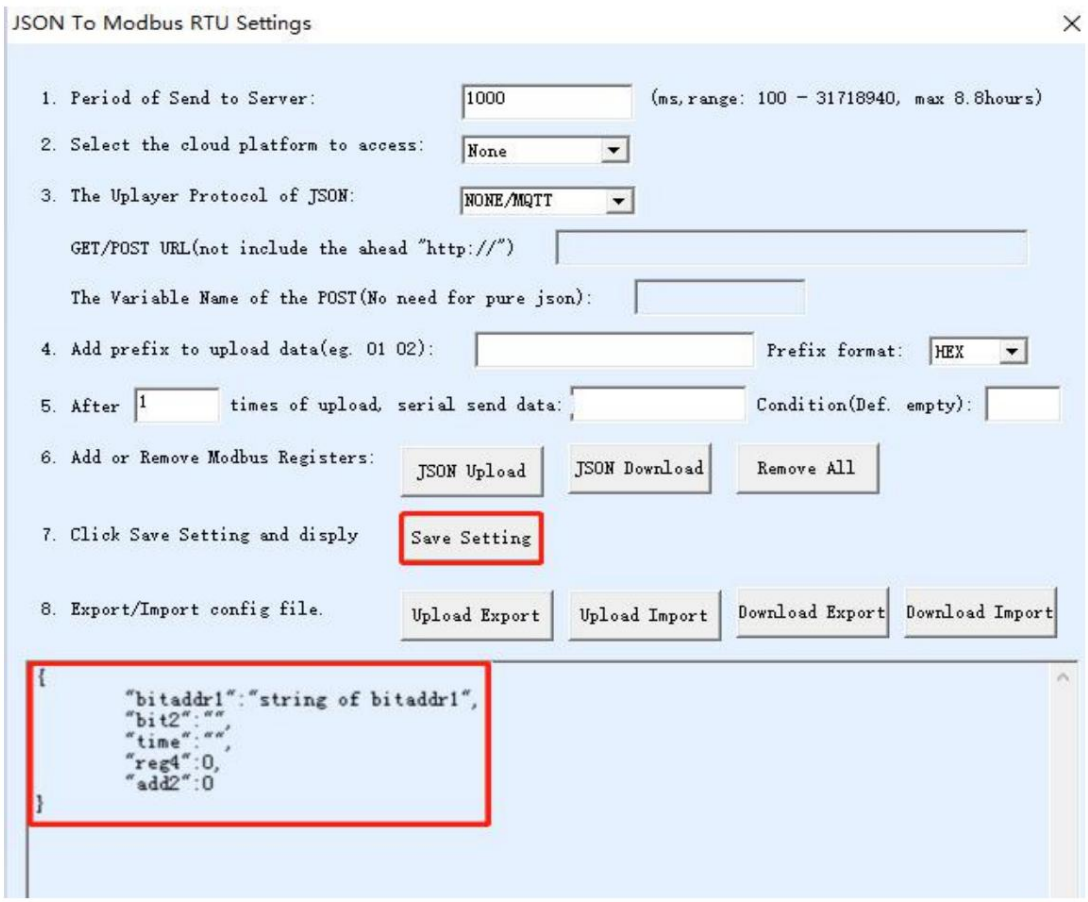
nie można wybrać liczby zarezerwowanych miejsc po przecinku.

3.5. POKAŻ WYNIKI PROJEKTOWANIA

Teraz kliknij „Zapisz projekt JSON”, aby zaprojektować i zobaczyć zawartość zaprojektowanego

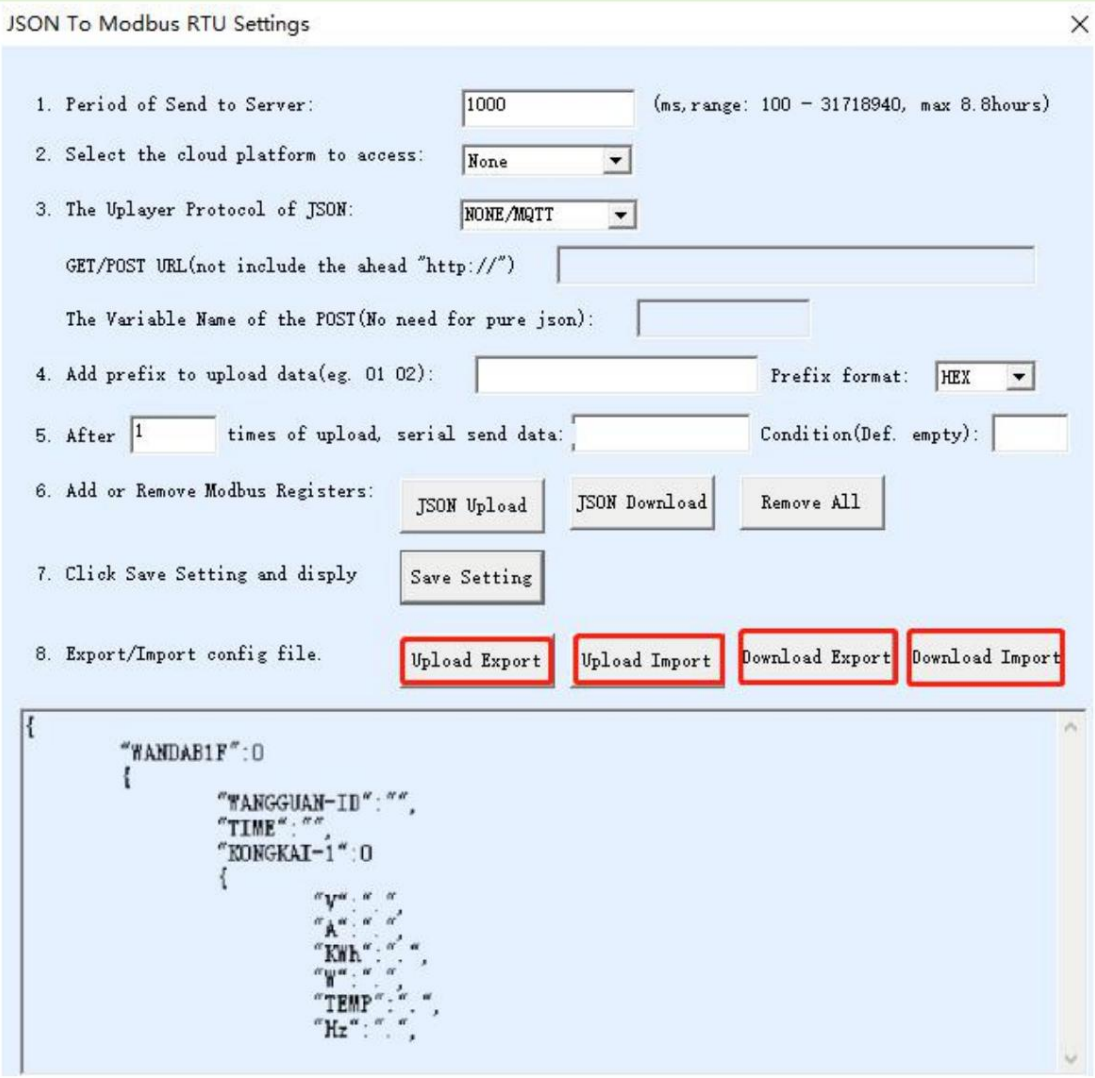
Format JSON w polu ekspozycyjnym, który jest wygodny do przeglądu projektu. W tym samym

czasie, przy wejściu w „Dodaj/Wyświetl” dostępny jest indeks umożliwiający porównanie.



Rysunek 11 przedstawia wyniki

3.6. EDYCJA W EXCELU



Rysunek 12 Import i eksport w formacie CSV

Dla ułatwienia edycji istnieje możliwość wyeksportowania zawartości projektu do formatu CSV, następnie edytuj za pomocą EXCELL, następnie zapisz jako CSV i zaimportuj.

Jednak CSV ma problemy z formatowaniem:

Num	JSON Keywords	ata Sourcesixed strin	s slave ado	s function	s register ado	45 device ID	45 data	ta Forma	byte ord	te Forma
1	CW	Modbus RTU	1	3	0	1	9410	2	0	gned inte
2	BW	Modbus RTU	1	3	1	1	9410	2	0	gned inte
3	BV	Modbus RTU	1	3	100	1	9410	2	0	gned inte
4	AV	Modbus RTU	1	3	101	1	9410	2	0	gned inte
5	CV	Modbus RTU	1	3	102	1	9410	2	0	gned inte

Rysunek 13 Błąd formatu danych

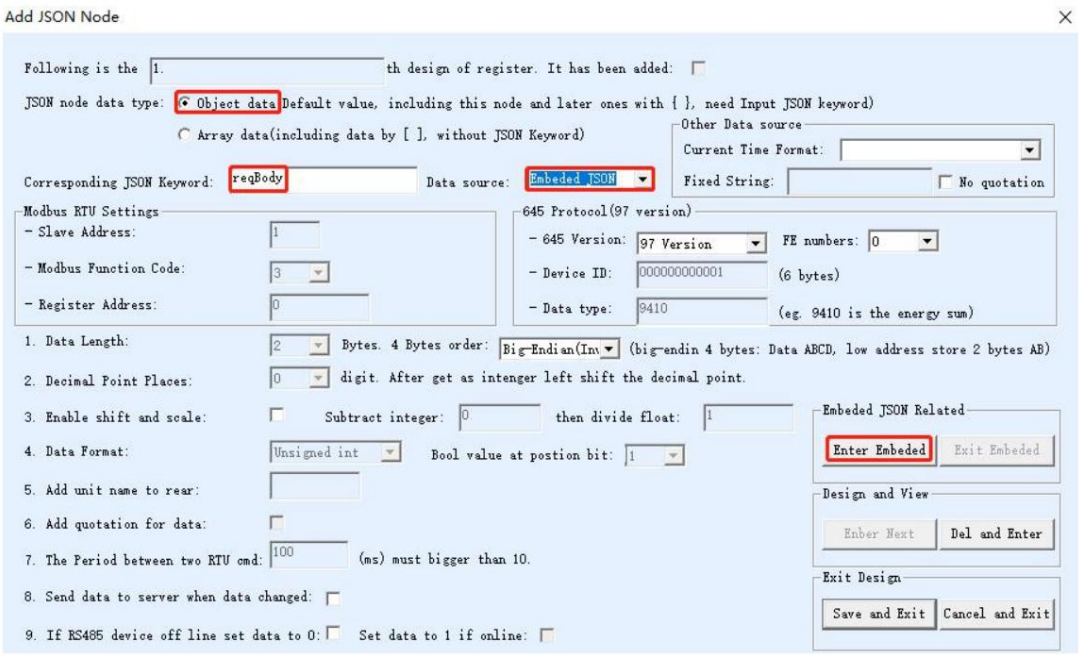
W tej chwili możesz zapisać plik CSV w formacie XLS do edycji. Po edycji zapisz jako CSV formatować i importować.

3.7. WSPORNIKI I ZESTAWY

Nowa wersja ZLVircom obsługuje projektowanie tablic JSON. Podaj prosty przykład:

```
{
  „reqBody”:
  [
        0,1
  ]
}
```

W tym przykładzie istnieje tylko jeden obiekt JSON reqBody, jego zawartością jest tablica, pierwsza elementem tablicy są dane 0, a drugim elementem są dane 1. W tym przypadku tablicę w nawiasach traktujemy podobnie jak zagnieżdżony JSON, ale ten zagnieżdżony JSON nie wymaga nazw słów kluczowych i dwukropków. Etapy projektowania są następujące:

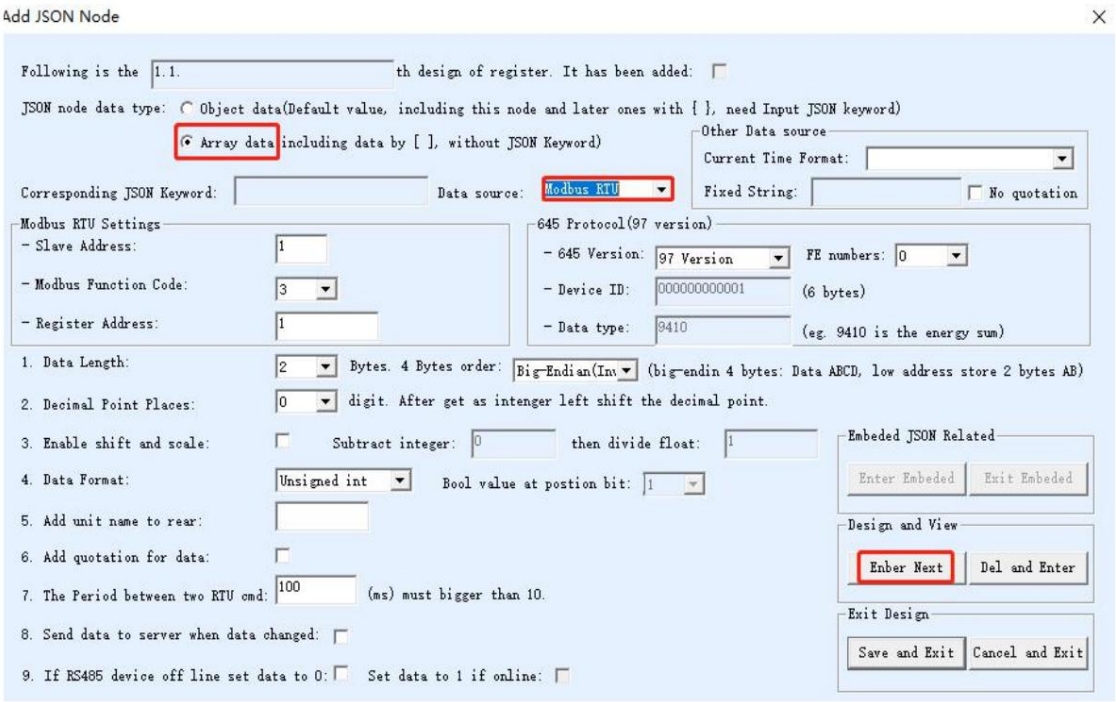


Rysunek 14 Obiekt tablicowy

Ponieważ samo reqBody jest obiektem, a nie jednostką tablicy, typ węzła jest wybierany jako obiekt

Dane, a nie dane tablicowe. Ponieważ jego zawartość jest tablicą, zgodnie z nawiasami są to nawiasy klamrowe

Pomysł jest taki, że źródłem danych tego reqBody jest „zagnieżdżony JSON”. Następnie kliknij „Projektuj zagnieżdżony”. JSON”.



Rysunek 15 Zawartość tablicy

Następnie zaprojektuj pierwszą zawartość tablicy, ponieważ jest to zawartość tablicy. Więc typem węzła są dane tablicowe. Źródłem danych jest zbiór Modbus RTU, wypełnij odpowiednie Rejestry Modbus i inne parametry. Następnie kliknij „Przejdź do następnego”, aby zaprojektować drugi element tablicy, a metoda jest podobna. Po zaprojektowaniu drugiego elementu od nie ma już projektu, kliknij „Zapisz wszystko i wyjdź”. Wróć do poprzedniego interfejsu i kliknij „Zapisz ustawienia JSON”, a następnie pobierz je. Zamieszczone dane to: {"reqBody":[2,3]}.

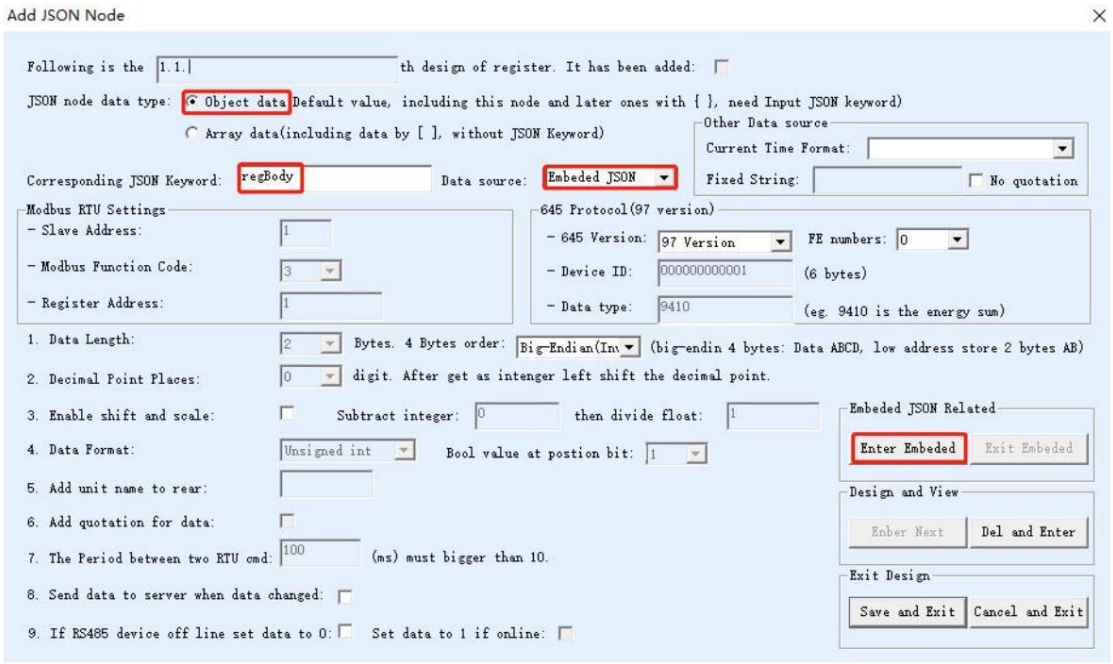
Spójrzmy teraz na bardziej skomplikowany przykład:

```
{
  „reqBody”:
  [
        {
            "id_warsztatu:"1008",
            „kod_maszyny”: „XS114”

        },{
             "id_warsztatu:"1008",
             „kod_maszyny”: „XS116”
         }
```

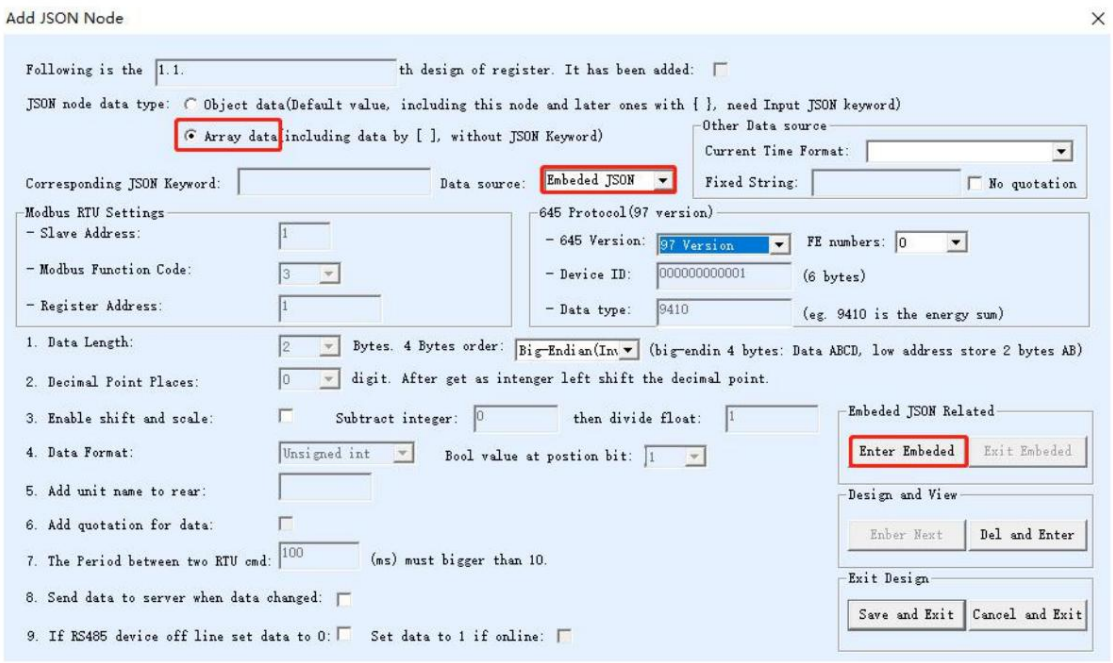
```
    ]
}
```

Zawartość tablicy nie jest tutaj prostą wartością danych, jest to sam JSON.



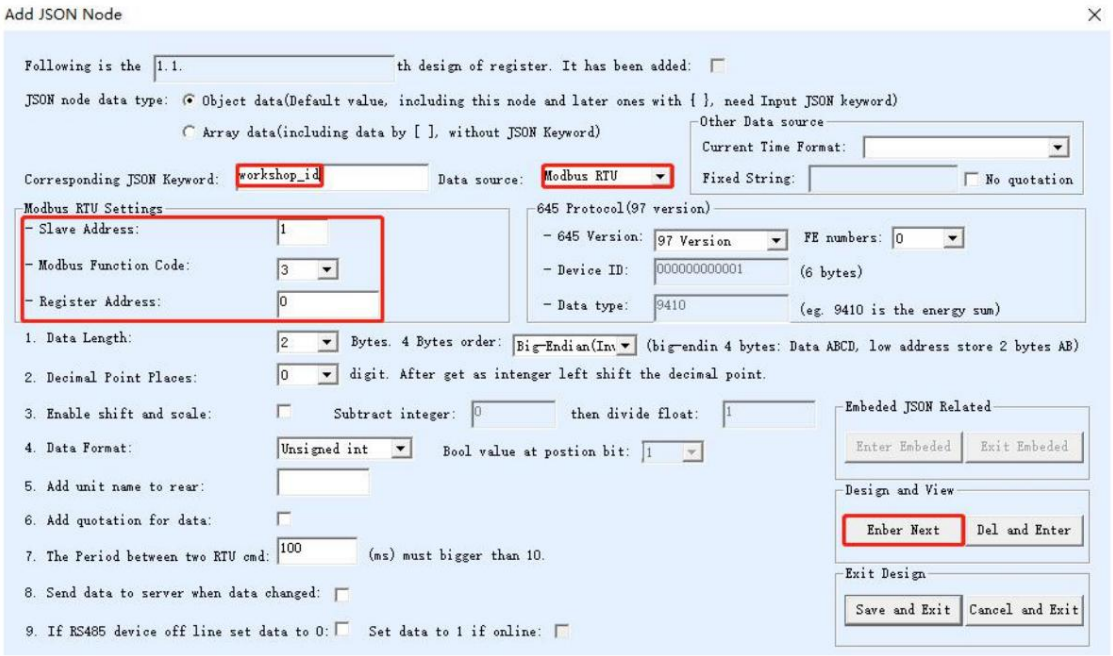
Rysunek 16 Krok 1

W wersji 1.1 poprzedni przykład 1 polega na wybraniu typu węzła jako „danych tablicowych”, a następnie bezpośrednio zaprojektuj dane w formacie Modbus, ale zawartość danych jest tutaj obiektem JSON, więc musisz wybrać źródło danych jako „zagnieżdżony JSON”, a następnie kliknąć „Zaprojektuj zagnieżdżony JSON”.

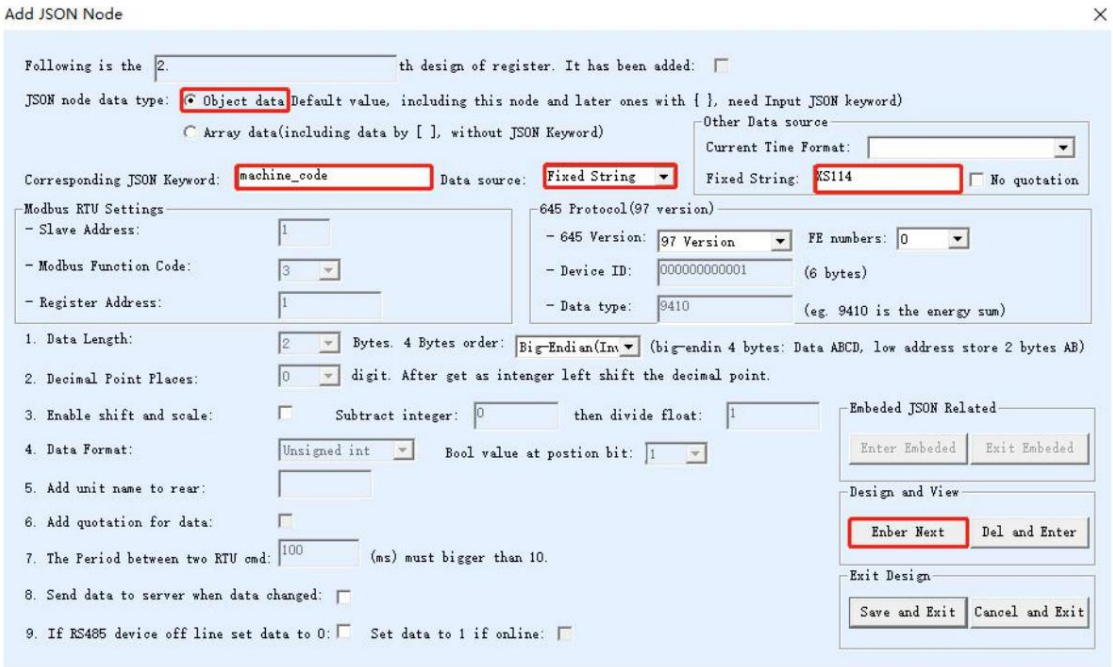


Rysunek 17 Krok 1.1

Następnie zaprojektuj dwa typy obiektów w krokach 1.1.1 i 1.1.2, dane źródłowe Modbus warsztat_id i stały ciąg danych źródłowych machine_code.



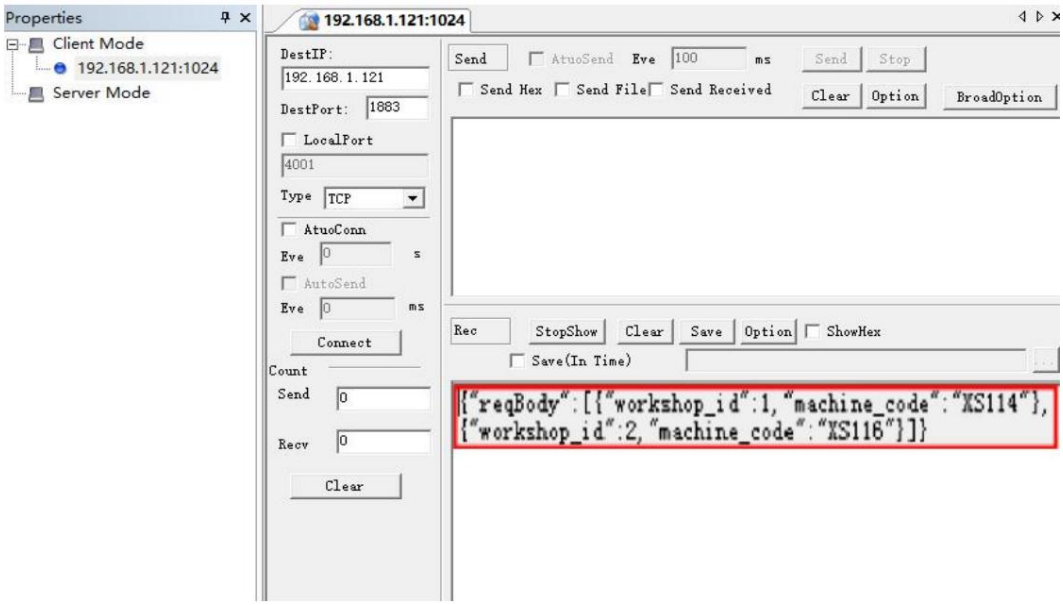
Rysunek 18 Krok 1.1.1



Rysunek 19 Krok 1.1.2

Pamiętaj, że musisz kliknąć „Przejdź do następnego” w pustym węźle treści (tzn. 1.1.3), kliknij „Powrót do poprzedniego poziomu”, aby przejść do „1.2”. 1.1.3 tutaj właściwie nie istnieje węzeł."1.2" jest również typem tablicy, a źródłem danych jest zagnieżdżony JSON, patrz krok 1.1. Następnie, aż do „1.2.2”, kliknij bezpośrednio „Zapisz wszystko i wyjdź”.

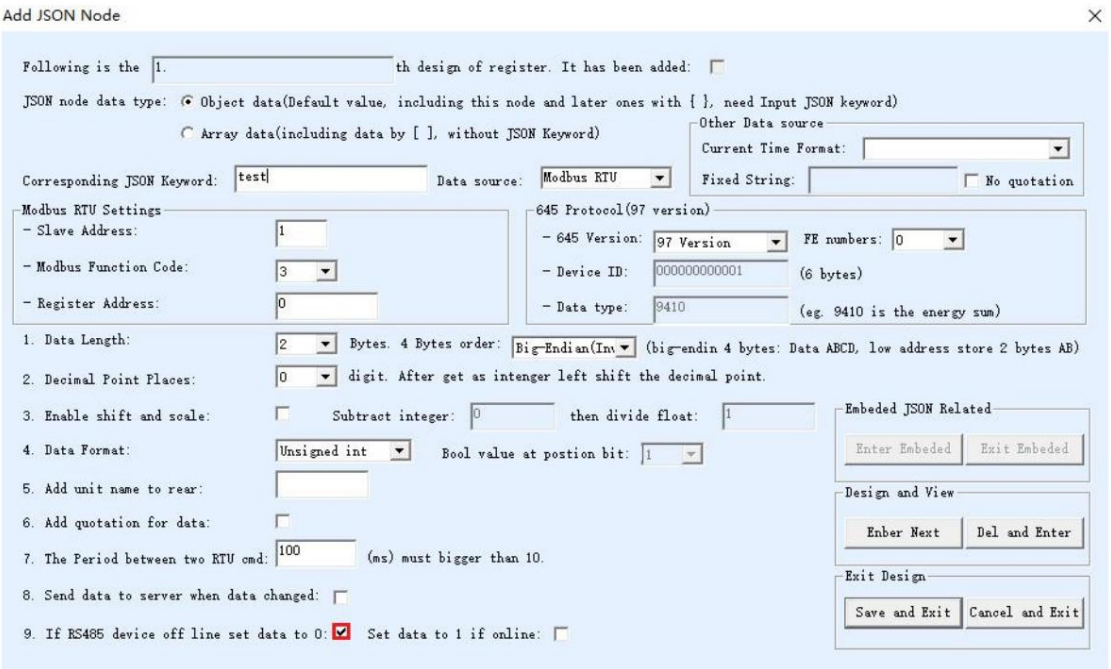
Format ostatnio przesłanych danych jest następujący, wśród których znajduje się identyfikator warsztatu odczytać z rejestru Modbus.



Rysunek 21 Krok 1.1.3

3.8. NIE MOŻNA CZYTAĆ I WYCZYŚĆ

Jeżeli nie można odczytać rejestru, można zastosować wartość danych 0, aby wskazać, że dane nie został przeczytany.

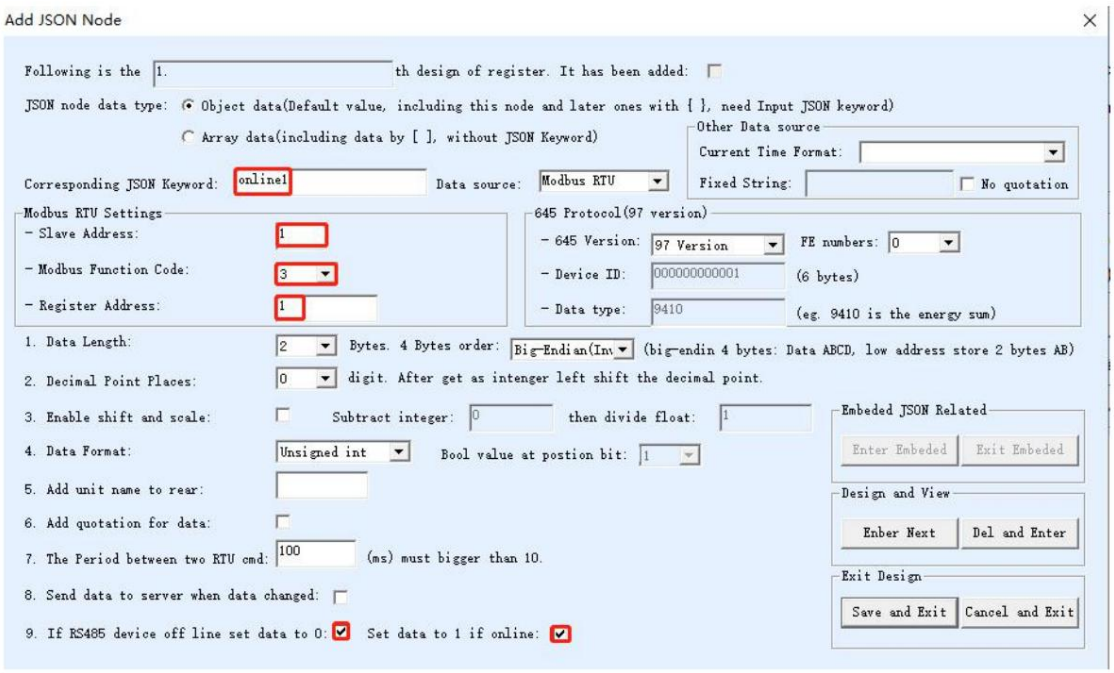


Rysunek 22 Wyczyść dane

Tutaj projektujemy testowy JSON, zwróć uwagę na sprawdzenie danych offline urządzenia RS485 clearing. Gdy rejestr może zostać odczytany, wysyłane są dane w postaci {"test":123}, gdzie 123 to rzeczywista wartość zarejestruj zawartość. Gdy urządzenie będzie w trybie offline lub nie będzie można odczytać danych, zostaną one przesłane jako {"test":0}. W ten sposób można uniknąć stwierdzenia, że dane nadal istnieją, gdy urządzenie jest w trybie offline, co może wprowadzić ludzi w błąd.

3.9. URZĄDZENIE W trybie offline

Jeśli danych nie można odczytać, dane wynoszą 0. W niektórych przypadkach nie można ocenić, czy urządzenie jest w trybie offline. Na przykład, jeśli sama zawartość danych wynosi 0, nie można ocenić, czy urządzenie jest w trybie offline lub dane mają wartość 0. Ponadto czasami urządzenie ma wiele rejestrów, które należy odczytać, dlatego należy użyć osobnego słowa kluczowego JSON, np. {online1:0} jako 0 lub 1, aby wskazać, że urządzenie jest online. W tym celu możesz zaprojektować online1 w następujący sposób. Najpierw zaprojektuj wszystkie rejestry, a następnie dodaj węzeł JSON:



Rysunek 23 Urządzenie online

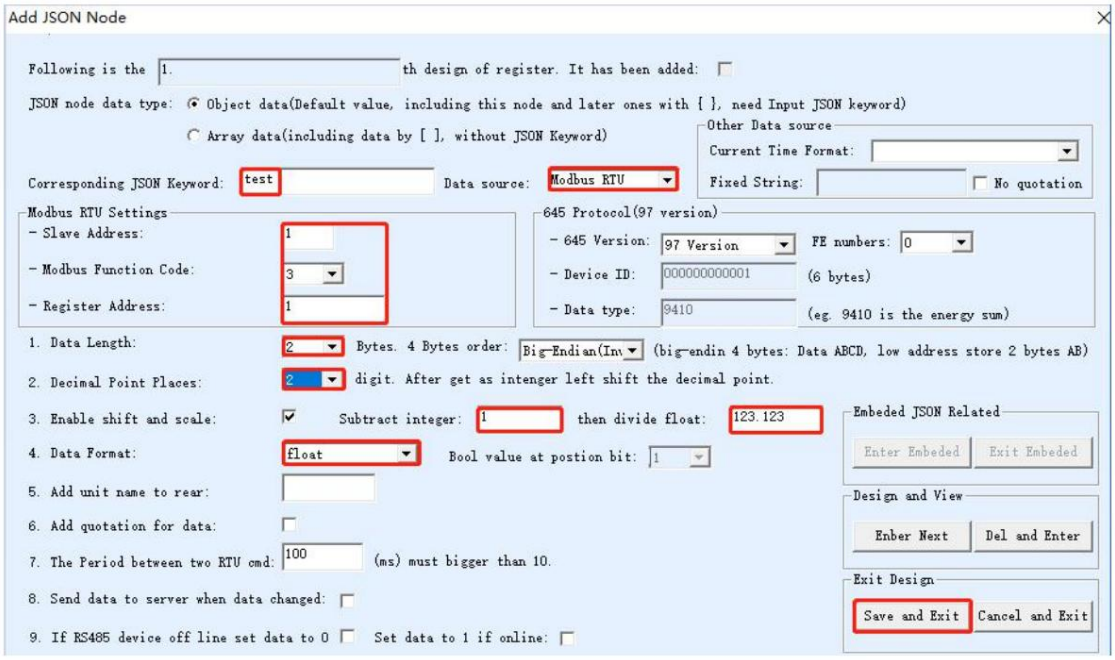
Nazwa tutaj to online1, którą można dowolnie przyjmować. Numer rejestru może wybrać dowolny istniejący rejestr. Najważniejszą rzeczą jest sprawdzenie danych offline urządzenia RS485 kasowanie i jeśli urządzenie jest w trybie online, niezależnie od zawartości rejestru, zostaje na siłę ustawione na 1. In w ten sposób, nawet jeśli odczytane dane wyniosą 0, będzie to 1 tak długo, jak będą odczytywane dane. Oznacza to, że online1 ma tylko dwie dane, 0 i 1. 1 oznacza online, 0 oznacza offline.

3.10. PRZESUŃ I POWIĘKSZ

Pan i Zoom to zawartość danych odczytywana przez rejestr Modbus minus 2-bajtowa liczba całkowita,

a następnie podziel przez liczbę zmiennoprzecinkową o pojedynczej precyzji, aby uzyskać liczbę zmiennoprzecinkową o pojedynczej precyzji

liczba zmiennoprzecinkowa, jak pokazano na poniższym rysunku:



Rysunek 24 Przesuwanie i powiększanie

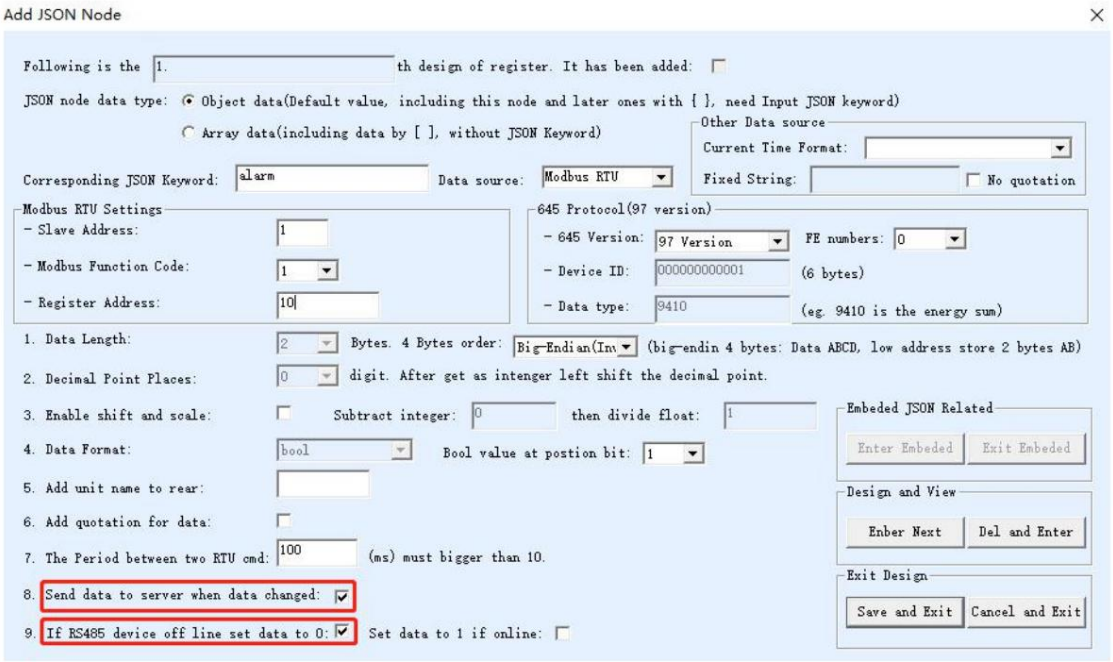
Jak pokazano na rysunku, zaprojektuj test słów kluczowych JSON, odczytaj adres stacji 1, 1 rejestr (2 bajtów) rejestru 0, następnie odejmij 1 i podziel przez 123.123. Jest to równoznaczne z przesunięciem danych odczytanych przez rejestr w dół o 2 jednostki, a następnie zmniejszając je 123,123 razy. Kiedy zawartość rejestru to 124, przesłane dane to {"test":0.99"}, (124-1)/123.123=0.99.

Należy pamiętać, że obsługuje to tylko źródło danych Modbus RTU, długość danych wynosi 2, 03 lub 04 kod funkcji, wyjście zmiennoprzecinkowe. Można wybrać liczbę miejsc dziesiętnych od 0 do 4. Najważniejszą rzeczą jest zaznaczenie opcji umożliwiającej włączenie panoramowania i powiększania. Wpisz liczba całkowita w minus całkowitej, zakres wynosi od -32768 do 32767, dzielona przez liczbę zmiennoprzecinkową Wprowadź a liczba zmiennoprzecinkowa w punktach.

3.11. RAPORTOWANIE ZMIAN DANYCH

W niektórych przypadkach, aby ograniczyć ruch danych, nie jest konieczne przesyłanie danych często. Wystarczy przesłać zebrane dane, gdy nastąpi zmiana. Obecnie, ZLVircom w wersji 5.30 i nowszej może realizować tę funkcję z oprogramowaniem 1.589. Aktualny Metoda polega na ustawieniu bardzo długiego okresu przesyłania, maksymalny wynosi 8,8 godziny. Z z punktu widzenia ruchu drogowego jest to równoznaczne z nie przesyłaniem danych w tym momencie, ale jeśli to sprawdzisz przesyłanie zmian danych, za każdym razem, gdy zmienią się dane węzła, spowoduje to przesłanie wszystkich dane. Zmianę danych można tutaj skonfigurować dla każdego węzła JSON i można do tego przywyknąć

wyzwalać przesyłanie, gdy urządzenie jest w trybie offline (w tym momencie zawartość danych zmienia się, więc może również zostać przesłane). W typowym przypadku zbieramy status DI i przesyłamy go, gdy się zmienia.



Rysunek 25 Przesyłanie zmian danych

Tutaj musisz sprawdzić zmianę danych, aby przesłać dane. Jeśli urządzenie RS485 jest w trybie offline reset, uruchomi raport, gdy urządzenie będzie w trybie offline. Jednak ten raport offline jest ograniczony do oryginalnej wartości danych wynoszącej 1. Jeśli wymagany jest jakikolwiek raport offline, oddzielny węzeł JSON trzeba dodać. Zapoznaj się ze wstępem w sekcji „Urządzenie w trybie offline”.

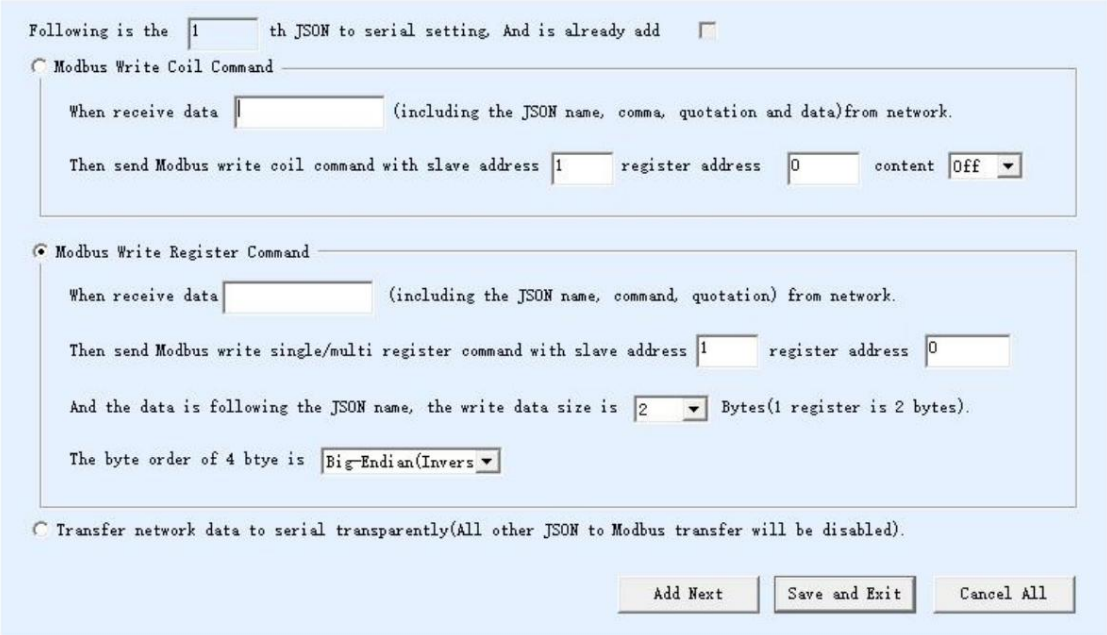
Dodatkowo dodaj kolejny węzeł alarm2 o adresie 11, ale odznacz zmianę danych wgrywać.

Normalne dane są przesyłane raz na 10 sekund. Jeżeli bit adresu 10 ulegnie zmianie (alarm) o godz tym razem raport zostanie uruchomiony natychmiast (właściwie zajmuje to kilkaset milisekund, aby dane uległy zmianie w związku z rotacją, a przesyłanie będzie niewielkie opóźniony). Jeśli jednak bit adresu 11 ulegnie zmianie, raport nie zostanie wygenerowany, ponieważ opcja raportu zmian nie jest zaznaczona.

Gdy alarm ma wartość 1, raport zostanie uruchomiony natychmiast, jeśli urządzenie będzie w trybie offline.

3.12. WYDANIE JSON

Emisja JSON realizuje dopasowanie pojedynczego słowa kluczowego JSON i wyniku pliku odpowiednie polecenie Modubs. Nie wymaga to, aby wydany ciąg był zgodny dokładnie, ale o ile w wydanym pliku znajdują się odpowiednie słowa kluczowe JSON i powiązane dane dane.



Rysunek 26 Wysyłanie JSON do Modbus

Kliknij przycisk „Pobieranie JSON” w „Ustawieniach JSON do Modbus RTU”, aby otworzyć plik powyżej okna dialogowego. Jest on podzielony na trzy kategorie: instrukcje cewek nastawczych Modbus, instrukcje rejestru zapisu Modbus i przejrzyste instrukcje transmisji.

Polecenie cewki nastawczej Modbus: Jest to polecenie 05, w którym można określić urządzenie podrzędne adres, zarejestruj adres i ustaw go na Włączony i Wyłączony. Pod względem danych, jeśli odpowiadają one JSON „alarm”: „1”, musisz wpisać w polu cały „alarm”: „1””, włączając cudzysłowy i dwukropki.

Instrukcja zapisu rejestru Modbus: jest to instrukcja 06, obecnie obsługuje od 2 do 4 bajtów, czyli zapisuje 1 i 2 rejestry. Dane muszą być liczbą całkowitą (bez przecinka dziesiętnego). Tutaj możesz ustawić adres stacji wysyłającej, adres rejestru i liczbę bajtów. Jeśli jest to 4 bajtów, możesz wybrać format big-endian lub format small-endian. Dla wygody danych, jeśli odpowiednią metodą jest „temp”: „1234”, należy wprowadzić „„temp”: „” na wejściu skrzynka. Tutaj należy wpisać znaki przed 1234, łącznie ze znakami cudzysłowu.

W przypadku metody transmisji przezroczystej wydane polecenie będzie transparentne przesyłane do portu szeregowego bez analizy. Po wybraniu strony przekazującej

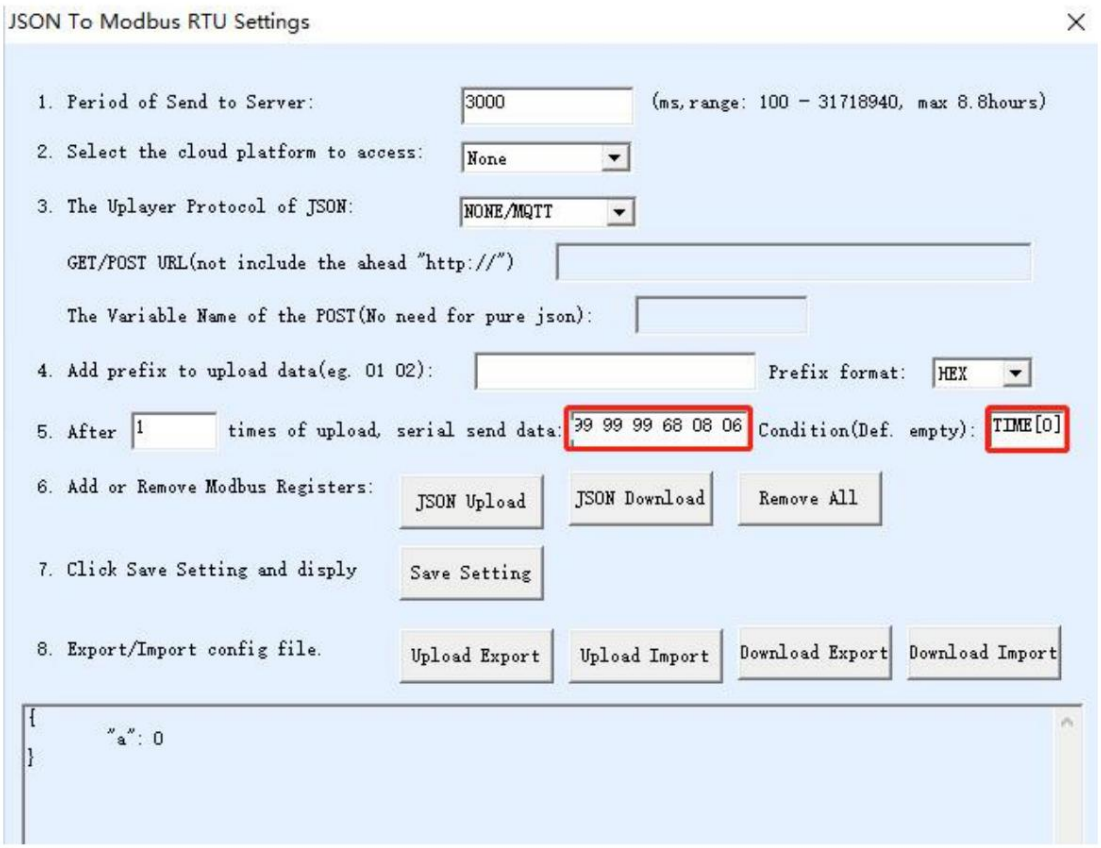
W ten sposób całe dostarczanie i analizowanie JSON zostanie wyłączone.

3.13. 645 TERMIN ZAWARCIA UMOWY

Usługa czasu protokołu 645 wynosi 68 99 99 99 99 99 99 68 08 06 SS MM HH DD MM YY CS 16, gdzie SS MM HH DD MM YY to sekundy, minuty, godziny, dni, miesiące i lata.

Współpracuj z oprogramowaniem ZLVircom5.32 i 4.99 (ZLSN7044)/5.92 (ZLSN2043), aby zrealizować konfiguracja rozrządu. Skopiuj następujący ciąg znaków „68 99 99 99 99 99 99 68 08 06 CZAS[25...30] CRC[3] 16" do portu szeregowego w celu jednoczesnego wysyłania poleceń wyjściowych; skopiuj „CZAS [0]” do warunku wyjściowego.

Pamiętaj, że aby zaimplementować synchronizację, musisz ustawić co najmniej jedno dane JSON do wysłania i dostawa. Ponadto obsługę czasu można wykonać dopiero po nawiązaniu połączenia TCP ustanowione, ponieważ usługa czasu następuje w momencie wysłania sieci. Jeśli TCP nie jest ustalony, nie zostanie wysłany i nie ma usługi czasu.



Rysunek 27 Konfiguracja rozrządu 645

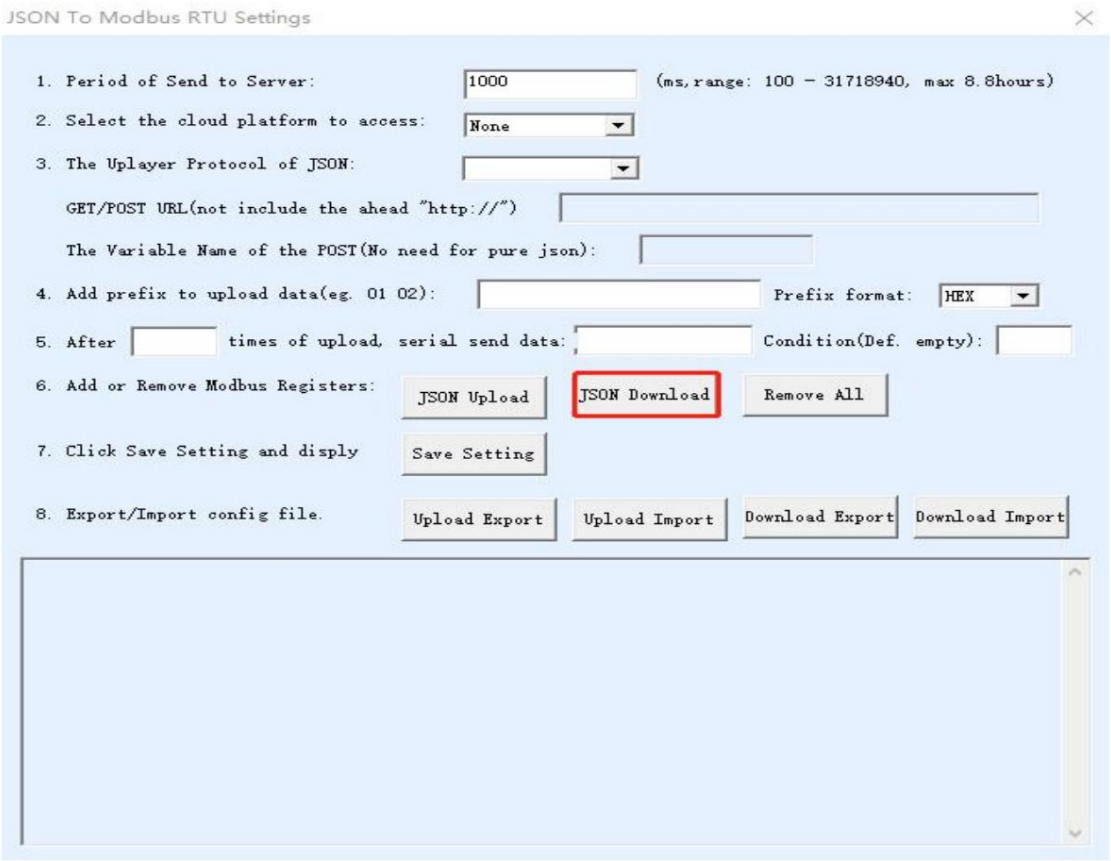
Wartość „CZAS[0]” jest tutaj opcjonalna. Jeżeli zostanie wypełniony, usługa czasowa zostanie wysłana po godz urządzenie uzyskuje efektywny czas. Jeżeli nie wypełnisz, usługa czasowa zostanie wysłana pod dowolnym okoliczności. Jeśli usługa czasu jest nieprawidłowa, spowoduje to problemy.

4. JSON DO MODBUS RTU

JSON do Modbus RTU obsługuje polecenie 05/06/16. Jeśli chcesz użyć polecenia 15 aby ustawić wiele cewek, użyj wielokrotnie polecenia 05.

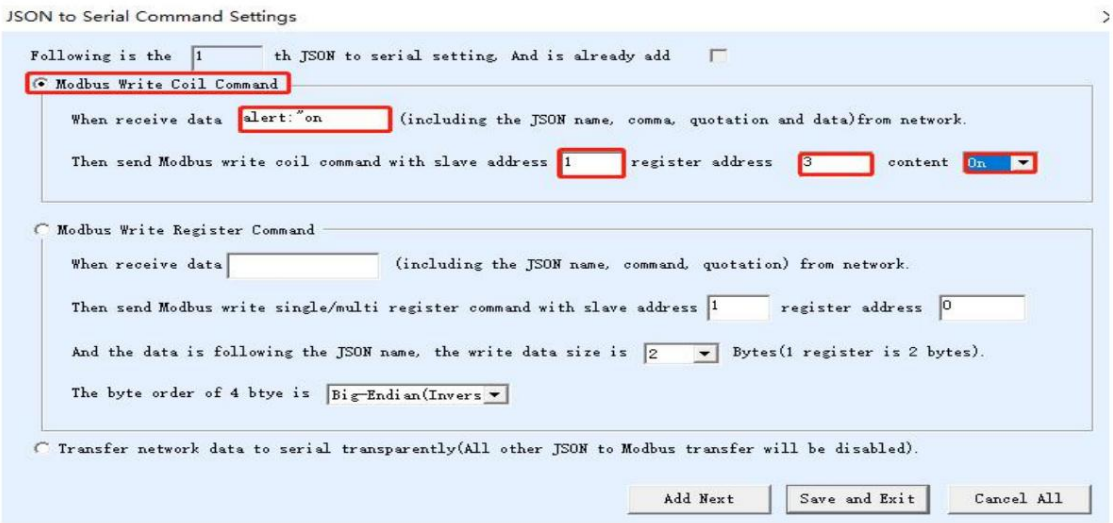
W zależności od długości liczby bajtów, system automatycznie wybierze Polecenie 06 lub 16 do wysłania. Oto przykłady ustawiania cewek i ustawiania rejestrów.

Jeśli otrzymasz dane JSON {alert:"on"}, musisz użyć polecenia 05, aby ustawić adres stacji 02 i cewka zaczynająca się od rejestru 03. Następnie w: JSON do interfejsu Modbus, kliknij „Dostawa JSON”



Rysunek 28. Wpisz JSON do wysłania

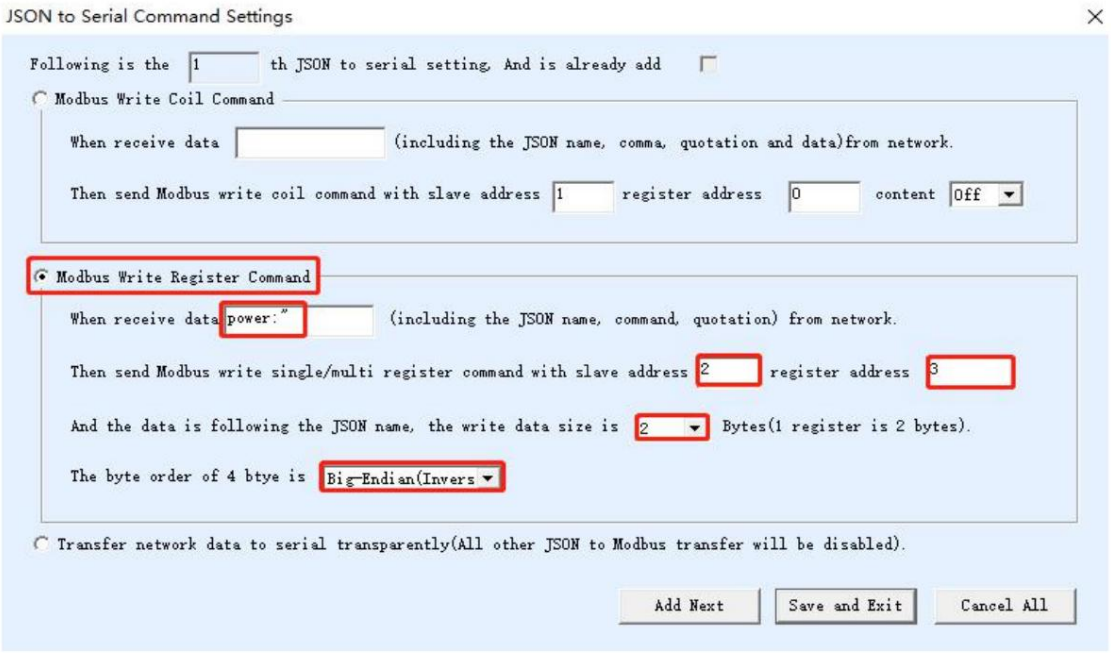
Interfejs konfiguracyjny wygląda następująco: zwróć uwagę na alert: „po ustawieniu tego trzeba napisać.



Rysunek 29 Konfiguracja cewki

Kliknij „Dalej”, aby dodać kolejną konwersję dostawy, w przeciwnym razie kliknij „Zapisz wszystko i wyjdź”. Po powrocie do głównego interfejsu kliknij „Zapisz ustawienia JSON”, a następnie kliknij „Wróć”. Następnie zwróć uwagę na kliknięcie „Pobierz” w interfejsie pobierania. To kończy konfiguracja.

Jeśli teraz zostanie wysłana moc:"12345"}, należy ustawić wartość mocy 12345 na stację adres 2 i rejestr 3. Ustawienia są następujące:



Rysunek 30 Rejestr ustawień JSON

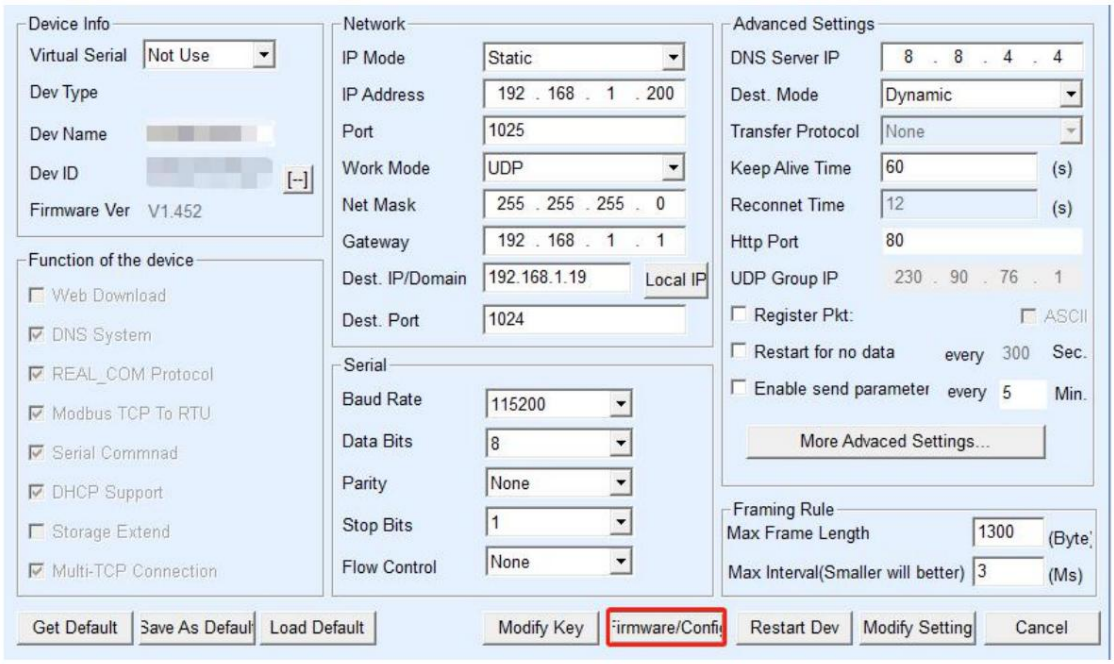
Pamiętaj, że w tym miejscu słowo kluczowe wymaga jedynie wprowadzenia potęgi:”, a Ty nie musisz tego wprowadzać następujący 12345, ponieważ ta wartość jest zmieniana, ale musisz wprowadzić dwukropek. Jeśli w wydawanych danych znajdują się cudzysłowy, należy je także wpisać w cudzysłów.

5. MQTT

MQTT może być używany samodzielnie lub w połączeniu z funkcją JSON. Używany samodzielnie, funkcja MQTT w przejrzysty sposób przesyła dane portu szeregowego do serwera MQTT. To jest dane odbierane przez port szeregowy są wykorzystywane jako obciążenie MQTT. Jednocześnie ładunek protokołu MQTT będzie wyprowadzany z portu szeregowego w przejrzysty sposób. Zrealizuj port szeregowy do MQTT.

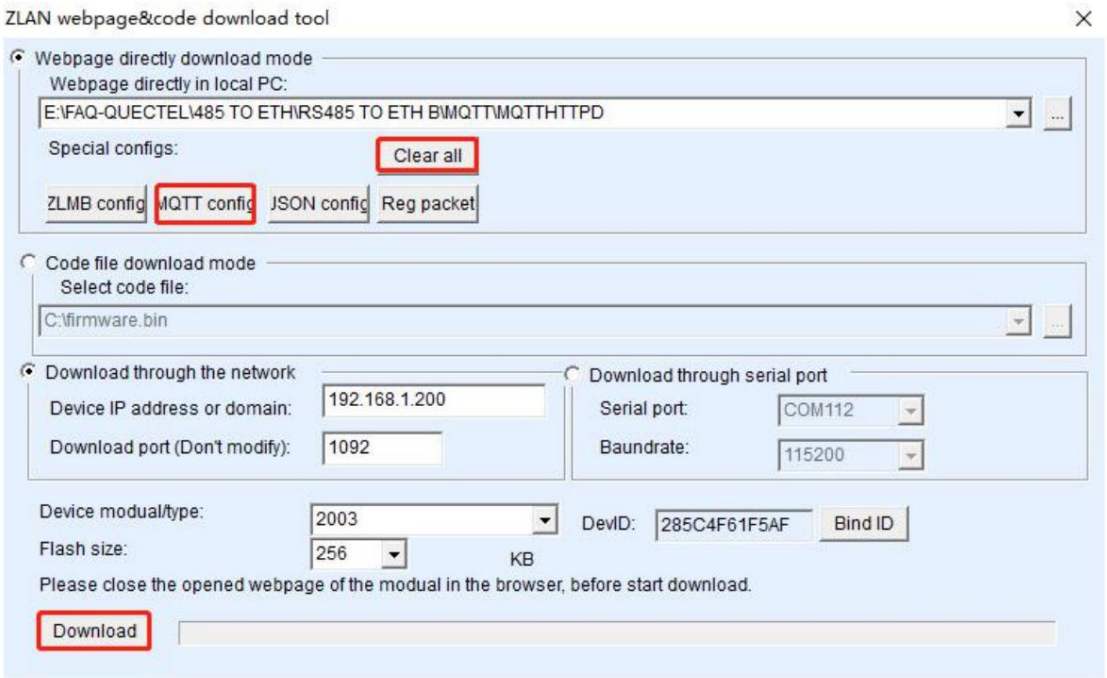
5.1. KONFIGURACJA URZADZENIA

Najpierw wyszukaj urządzenie, a następnie kliknij Edytuj urządzenie:



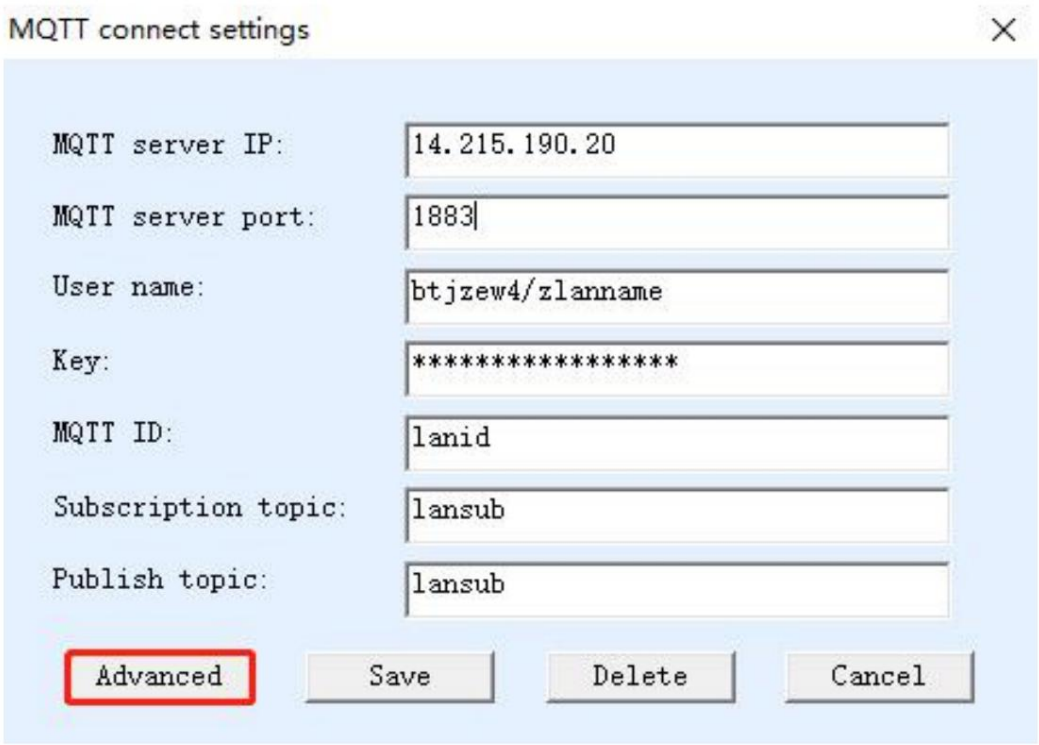
Rysunek 31 Konfiguracja MQTT 1

Kliknij „Oprogramowanie sprzętowe i konfiguracja”, okno dialogowe pobierania i projektowania konfiguracji pojawi się:



Rysunek 32 Konfiguracja MQTT 2

Tutaj wybierz „Pobieranie katalogu internetowego”, a następnie wybierz pusty katalog, np MQTTHTTPD, a następnie kliknij „Wyczyść wszystko”, aby wyczyścić poprzedni projekt (pamiętaj, że jeśli poprzedni projekt był zaprojektowany według JSON, nie usuwaj wszystkiego, w przeciwnym razie poprzedni projekt zostanie wyczyszczony. Projekt JSON). Następnie kliknij Konfiguracja MQTT.



Rysunek 33 Konfiguracja MQTT 3

Instrukcje konfiguracji są tutaj następujące:

1 Nazwa domeny serwera lub IP: tutaj jest adres IP serwera MQTT, maksymalna długość to 30 postacie.

2 Nazwa użytkownika: to nazwa użytkownika serwera MQTT.

3 Hasło: to hasło logowania tego użytkownika.

4 Identyfikator klienta: Jest to identyfikator klienta MQTT.

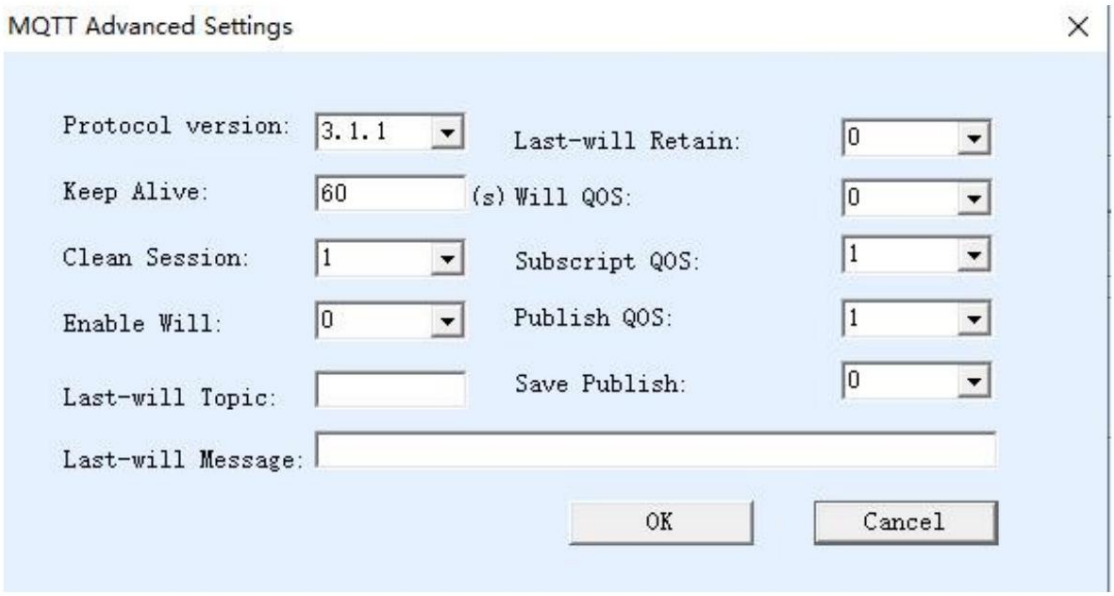
5 Subskrybuj temat: Jest to temat subskrybowany przez to urządzenie. Kiedy inne urządzenia opublikuj ten temat, serwer wyśle go na to urządzenie. Ogólnie rzecz biorąc, jeśli tylko publikujesz nie musisz wypełniać tego pola.

6 Publikuj temat: Temat danych wysyłanych do serwera po podłączeniu portu szeregowego urządzenia przekonwertowany na MQTT.

7 Zaawansowane parametry MQTT: używane do konfiguracji zaawansowanych parametrów.

8 Zapisz ustawienia MQTT: Po zaprojektowaniu kliknij ten przycisk, aby zapisać, a następnie kliknij „przycisk pobierania” w katalogu pobierania strony internetowej, aby pobrać.

Teraz kliknij „Zaawansowane parametry MQTT” (zwykle nie ma potrzeby konfigurowania zaawansowanych parametry):



Rysunek 34 Zaawansowana konfiguracja parametrów MQTT

Opisane w następujący sposób:

1 Wersja protokołu: Obecny główny nurt to wersja 3.1.1, jeśli chcesz wybrać wersję 3.1, wybierz tutaj.

2 Czas podtrzymania: Czas pulsu MQTT, minimum to 10 sekund, a wartość domyślna to 60 sekund.

3 Wyczyść subskrypcję serwera: czy serwer czyści później informacje o subskrypcji klient jest odłączony.

4 Czy włączyć ostatnie życzenie: czy istnieje ostatnie życzenie.

5 Motyw ostatniego życzenia: Motyw ostatniego życzenia.

6 Informacje o ostatniej woli: informacje o ostatniej woli.

7 Czy zapisać ostatnie życzenie: czy serwer musi zachować wiadomość z ostatnim życzeniem wysyłane do klienta, gdy klient jest nienormalnie offline.

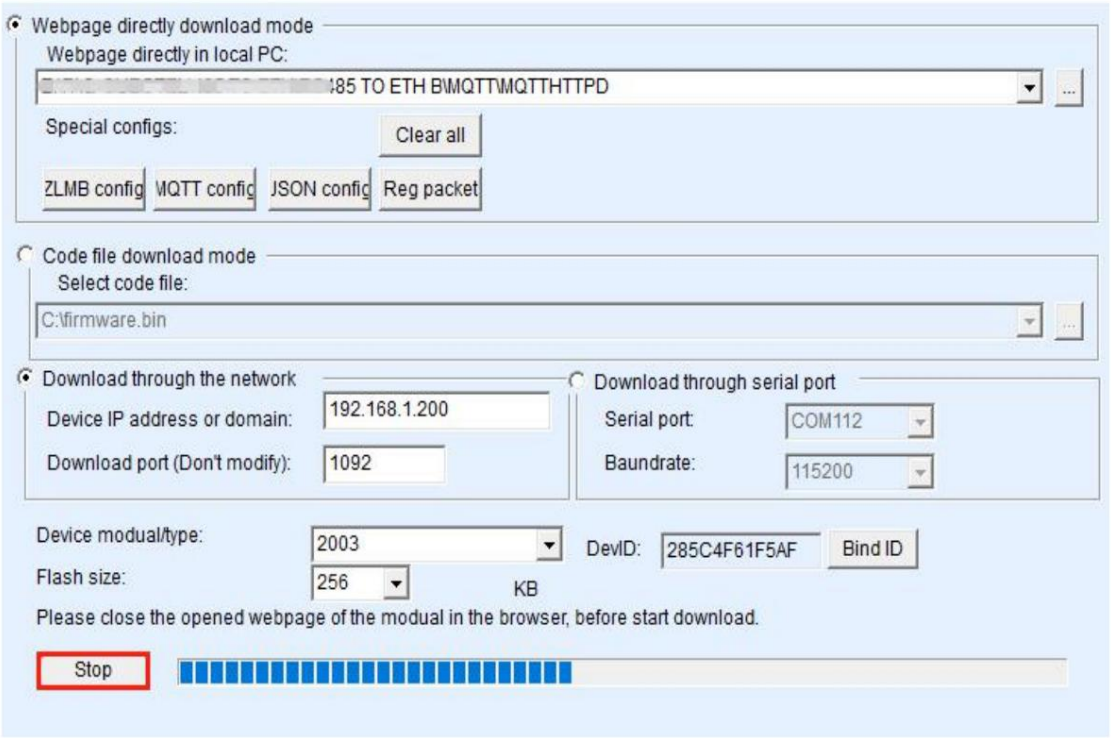
8 Jakość ostatniego życzenia: poziom jakości dostarczenia wiadomości ostatniego życzenia wysłanej przez serwer.

9 Jakość subskrypcji: poziom jakości dostarczania subskrypcji. W niektórych przypadkach to musi być ustawiony na 0, aby zapobiec rozłączeniu spowodowanemu retransmisją.

10 Jakość publikacji: Poziom jakości dostarczenia wiadomości opublikowanej przez klienta. W w niektórych przypadkach należy ustawić wartość 0, aby zapobiec rozłączeniu spowodowanemu retransmisją.

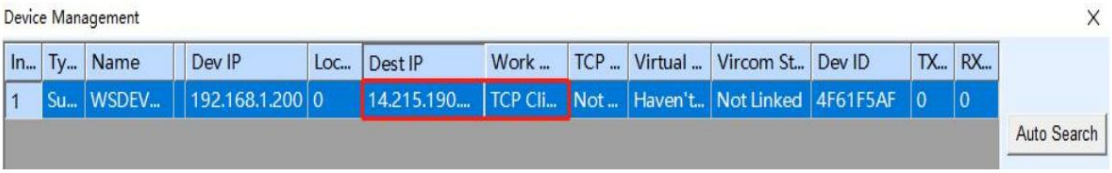
11 Czy zapisać publikację: czy serwer przechowuje ostatnią wiadomość (jeśli istnieje nowej subskrypcji klienta, zostanie ona wysłana do klienta)

Nie będziemy tutaj modyfikować parametrów zaawansowanych. Kliknij bezpośrednio „Zapisz ustawienia MQTT”. Następnie kliknij „pobierz”:



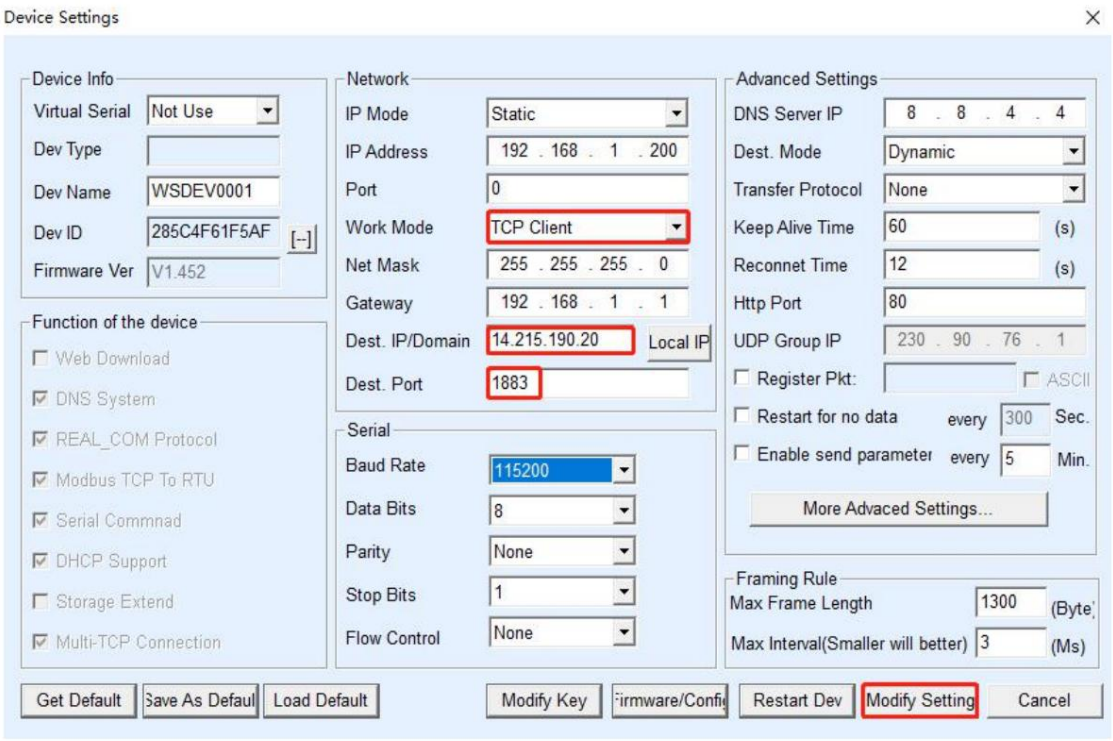
Rysunek 35 Pobieranie

Po pobraniu kliknij OK, a wrócisz do okna dialogowego zarządzania urządzeniami. Widać, że docelowy adres IP urządzenia, tryb pracy i port docelowy zostały ustawione automatycznie zmodyfikowane do ustawień MQTT:



Rysunek 36 Automatyczna modyfikacja

Jeśli nie ma automatycznej modyfikacji, należy ustawić docelowy adres IP, tryb pracy, i port docelowy w oknie dialogowym edycji urządzenia. Następnie kliknij „Edytuj ustawienia”.



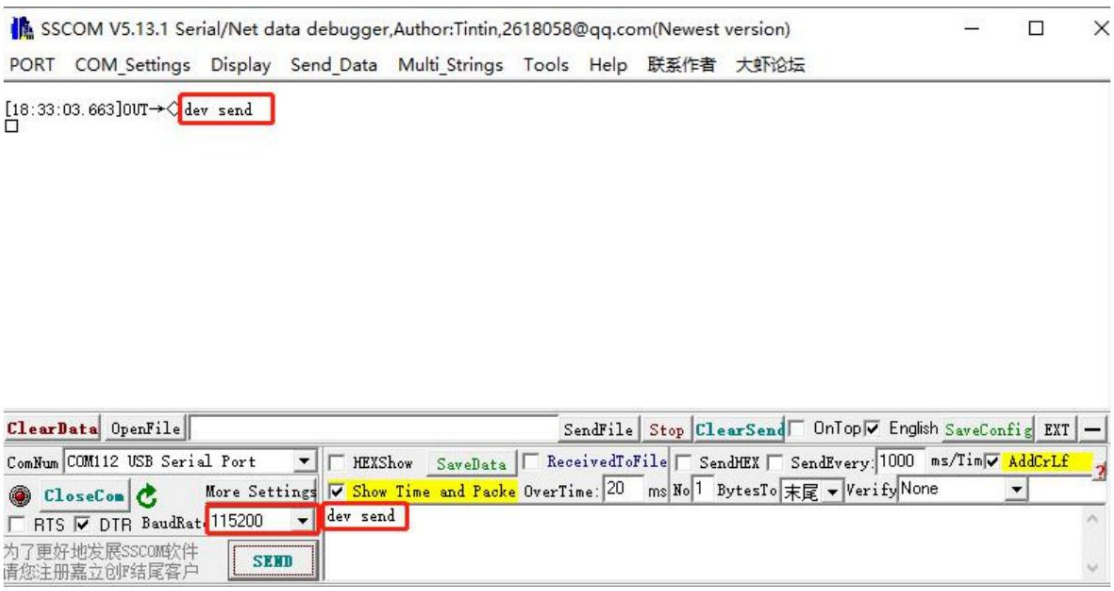
Rysunek 37 Konfiguracja IP

Ta konfiguracja została ukończona.

5.2. TEST DANYCH

Po nawiązaniu połączenia zaświeci się lampka LINK (zwykle niebieska dioda pośrodku). urządzenie włącza się. Wskazuje, że urządzenie jest normalnie połączone z serwerem MQTT.

Teraz otwórz narzędzie portu szeregowego:



Rysunek 38 Wysyłanie i odbieranie przez port szeregowy

Użyj tej samej szybkości transmisji co urządzenie, aby otworzyć port szeregowy i wysłać dane „dev send”, a następnie zobacz zwrócone dane „dev send” w oknie odbierającym. To dlatego, że my opublikuj wiadomość wysłaną przez dewelopera na serwer MQTT w temacie zlansub. Ale na w tym samym czasie nasze urządzenie jest również zasubskrybowane w temacie zlansub, więc serwer od razu to zrobi wyślij nam wiadomość subskrypcyjną, a treść wiadomości subskrypcyjnej zostanie wysłana przez dewelopera. Informacje te są wysyłane i pobierane jako ładunek MQTT i wyprowadzane z portu szeregowego port poprzez przezroczystą transmisję.

Jeśli inne urządzenia publikują informacje, to urządzenie może również odbierać dane.

Ogólnie rzecz biorąc, użytkownicy mogą bezpośrednio i przejrzyście przesyłać polecenia portu szeregowego (takich jak Modbus RTU) do serwera MQTT. Ponadto możesz także skorzystać z Funkcja JSON wykorzystująca automatyczny zbiór formatów Modbus RTU i zwykły format JSON przesyłanie. Ponadto można również znaleźć Shanghai ZLAN, aby dostosować niektóre niestandardowe instrumenty i formaty protokołów komputera hosta.

6. MQTT+JSON DO MODBUS RTU

Połączenie powyższego JSON i MQTT może osiągnąć następujące funkcje:

1. Do nawiązania połączenia z serwerem i danymi wykorzystywany jest protokół oparty na MQTT komunikacja odbywa się w formie prenumeraty i publikacji.
2. Wsparcie niezależnego projektowania i automatycznego gromadzenia rejestrów Modbus RTU.
3. Obsługuj konwersję określonej zawartości rejestru Modbus do formatu JSON i wyślij ją regularnie i aktywnie.
4. Obsługa dodawania identyfikatora urządzenia do formatu JSON, aby ułatwić identyfikację urządzeń w chmurze.

Jeśli potrzebujesz funkcji MQTT+JSON do Modbus RTU, możesz zaprojektować MQTT i JSON oddzielnie, w dowolnej kolejności. Nie klikaj przycisku „Przejrzysty projekt” po zaprojektowaniu typ. Po ukończeniu obu projektów kliknij jednocześnie przycisk „Pobierz”, aby pobrać zawartość urządzenia.

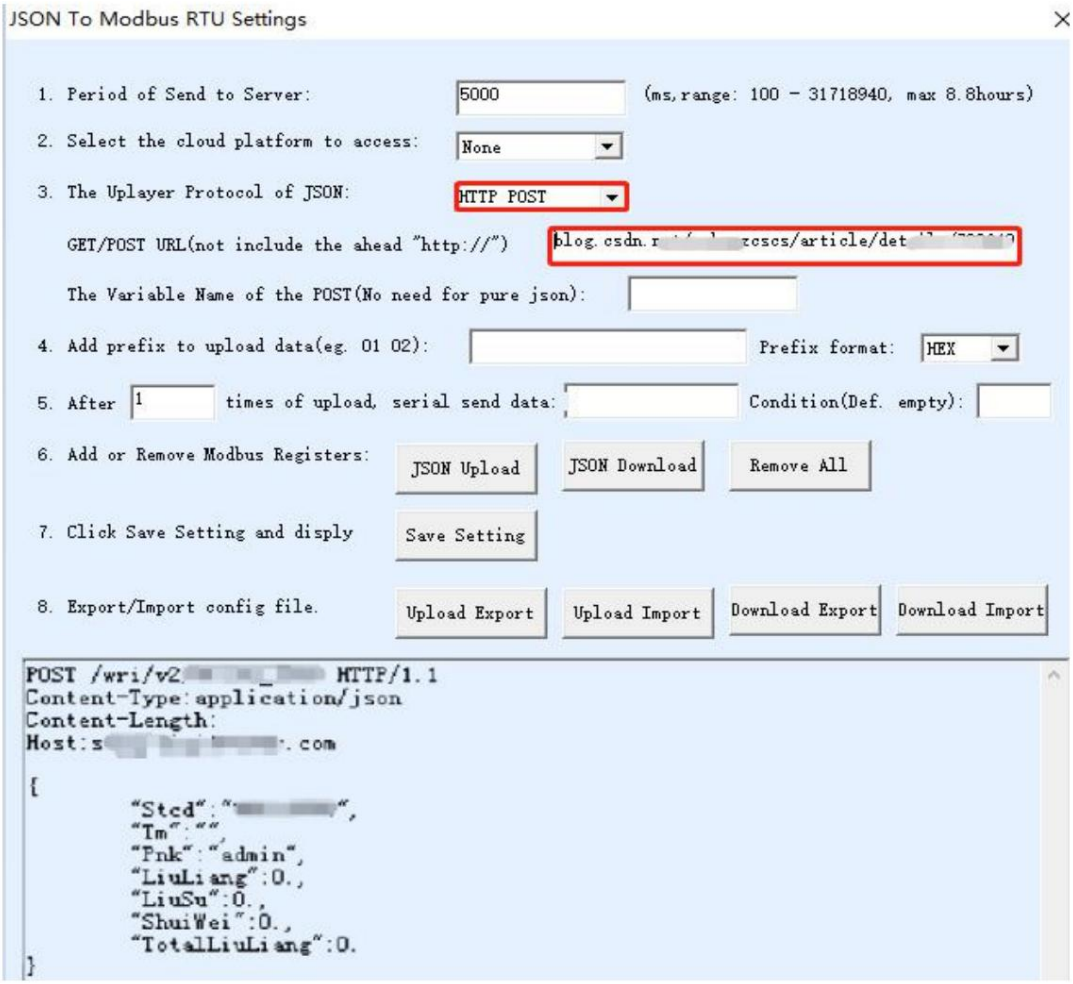
Ogólnie rzecz biorąc, po pobraniu można ręcznie ponownie uruchomić urządzenie, aby załadować ustawienia.

7. HTTP POST/GET+JSON

Oprócz MQTT protokół komputera hosta może również wybierać protokół HTTP i przesyłaj dane za pomocą instrukcji POST i GET. Przyjmijmy polecenie POST jako przykład do wprowadzenia.

Jeśli chcesz obsługiwać funkcję POST/GET+JSON, wybierz ZLVircom narzędzie konfiguracyjne do wyboru wersji 5.17 i wyższej; jeśli chcesz obsługiwać polecenie POST, oprogramowanie sprzętowe 2003 musi być w wersji 5.81 i nowszej (tylko

Jeśli obsługujesz GET, możesz użyć zwykłej wersji oprogramowania sprzętowego 2003, która obsługuje JSON-a).



Rysunek 39. POST+JSON

ZLVircom wersja 5.17 dodaje dwie opcje w ustawieniach JOSN do Modbus RTU, jak pokazano Na rysunku:

1. Protokół wyższej warstwy JSON: Jeśli nie jest to protokół lub protokół MQTT, wybierz pierwszy pozycja: „BRAK/MQTT”. Jeśli jest to HTTP POST, wybierz drugi element „HTTP POST”, jeśli tak jest HTTP GET, wybierz trzecią pozycję „HTTP GET”.

2. POST/GET URL: Wybierając POST lub GET, musisz wpisać adres URL. Na przykład, jeśli Adres URL to http://sacom/wri/v2, usuń http:// z przodu i bezpośrednio wpisz sacom/wri/v2.

Pozostałe metody projektowania struktur JSON są takie same jak metody wprowadzone wcześniej. Po kliknięciu przycisku „Zapisz ustawienia JSON”, jeśli wybrano POST/GET, format nagłówka HTTP informacje zostaną dodane przed danymi JSON w celu obsługi transmisji HTTP protokół.

Ta metoda projektowania POST/GET jest prosta i praktyczna, a także można ją przeprowadzić łatwo i szybko realizować transmisję Modbus RTU i innych danych przyrządu do serwera za pomocą środków protokołu HTTP POST/GET+JSON.