

Komunikacja sterownika Astraada One z wyspą Astraada IO po Modbus TCP

Komunikacja Modbus TCP dla AS70-R-MP-08DIO-P-M12 ze sterownikiem Astraada One

SPIS TREŚCI

Wstęp	1
Konfiguracja Modułu Astraada IO	2
Zmiana protokołu z wykorzystaniem programu Astraada IO Configuration Tool.....	2
Konfiguracja projektu w Codesys	4
Proces konfiguracji komunikacji Modbus TCP.....	4
Struktura przesyłanych danych	7
Potwierdzenie Wymiany danych i ich interpretacja.....	9

WSTĘP

W informatorze opisano krok po kroku sposób konfiguracji połączenia sterownika Astraada One z multiprotokołową wyspą Astraada IO. W tym przykładzie wykorzystano sterownik DC2000 (firmware 1.28.4), wyspę Astraada IO AS70-R-MP-08DIO-P-M12 oraz oprogramowanie Codesys V3.5 SP19 Patch 3.

KONFIGURACJA MODUŁU ASTRAADA IO

W celu uzyskania szczegółowych instrukcji można zwrócić się do podręcznika produktu dostępnego na stronie internetowej wsparcia firmy Astor ([Wsparcie -> Astraada -> Astraada IO](#)).

Fabryczne ustawienia modułu AS70-R-MP-08DIO-P-M12:

- Protokół: Profinet
- Adres IP: 192.168.0.2

Domyślny protokół tej jednostki to Profinet, dlatego należy skonfigurować jednostkę do pracy jako Modbus TCP slave oraz zmienić adres IP zgodnie z procesem opisanym poniżej:

Zmiana protokołu z wykorzystaniem programu Astraada IO Configuration Tool

Program, tak jak podręcznik produktu, jest dostępny do bezpłatnego pobrania na [stronie wsparcia Astor](#).

1. Uruchom program Astraada IO Configuration Tool,
2. Podaj obecny adres IP urządzenia, a w następnym oknie wpisz „p”, aby wybrać opcję zmiany protokołu (w przypadku zapomnienia adresu IP spójrz do podręcznika produktu na stronie wsparcia)
3. Wybór pozycji o numerze „4” spowoduje ustawienie protokołu Modbus TCP - postępuj zgodnie z poleceniami pojawiającymi się w terminalu i pamiętaj o resecie zasilania po zakończeniu konfiguracji.

```

/SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/
SS | _SS | /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/
SS | SS | /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/
SSSSSSSS SS \ SS | _SS | SS / SS / SS SS / SS / SS
SS | SS | /SSSSSS/ SS / SS | /SSSSSS/ /SSSSSS/ SS /SSSSSS/ /SSSSSS/
SS | SS | /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/
SS/ SS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/ /SSSSSS/

DEVELOPED BY JAHUB G., ASTOR | VERSION 1.7 | 05/03/2025

-----Please select language-----
[P] Polski | [E] English
Your selection | p

Podaj adres IP urządzenia: 192.168.5.4
Próba połączenia z 192.168.5.4
Nawiązano połączenie!

-----Wybierz jaki parametr chcesz zmienić-----
Wybierz [P], aby zmienić protokół
Wybierz [I], aby zmienić adres IP
Twój wybór | p

-----
Poniżej znajduje się lista dostępnych protokołów - wybierz wymagany protokół:
[1] EtherNet/IP | [3] CC-Link
[2] Profinet | [4] Modbus TCP
Twój wybór | 4
100%... Transmisja zakończona

Tylko dla modułów 8 punktowych! Proszę mieć na uwadze, że po zmianie protokołu modułowi zostanie nadany domyślny adres 192.168.0.2
Zamykanie połączenia z 192.168.5.4
Połączenie zamknięte

-----Konfiguracja zakończona - zresetuj zasilanie modułu-----
  
```

Uwaga! Po zmianie protokołu adres IP zostanie ustawiony na domyślny 192.168.0.2

4. Po resecie zasilania i zaświeceniu się diody Us na zielono uruchom program ponownie. Podaj **domyślny adres IP** urządzenia, a w następnym oknie wpisz „i”, aby wybrać opcję zmiany adresu IP
5. Postępuj zgodnie z poleceniami pojawiającymi się w terminalu i wprowadź następujące przykładowe ustawienia:

```
IP adres:      192.168.1.2
Maska podsieci: 255.255.255.0
Gateway:      192.168.1.1
```

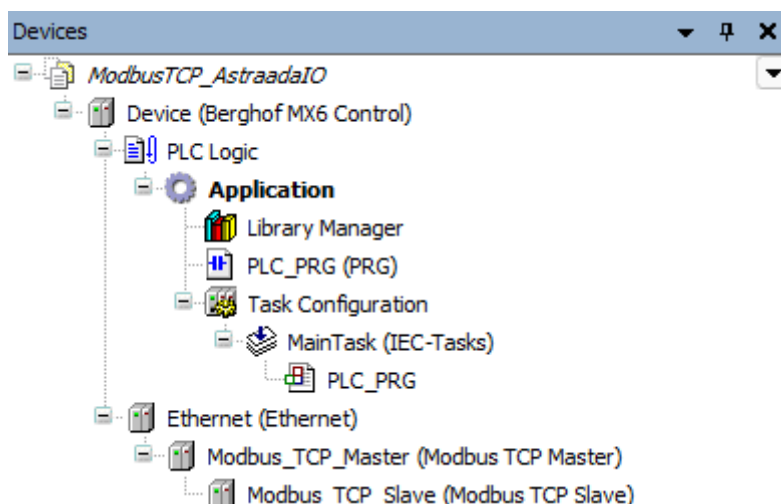
Pamiętaj o resecie zasilania po zakończeniu konfiguracji.

[illegible]

Po drugim resecie moduł jest gotowy do pracy jako Modbus TCP slave i można przystąpić do konfiguracji sterownika.

KONFIGURACJA PROJEKTU W CODESYS

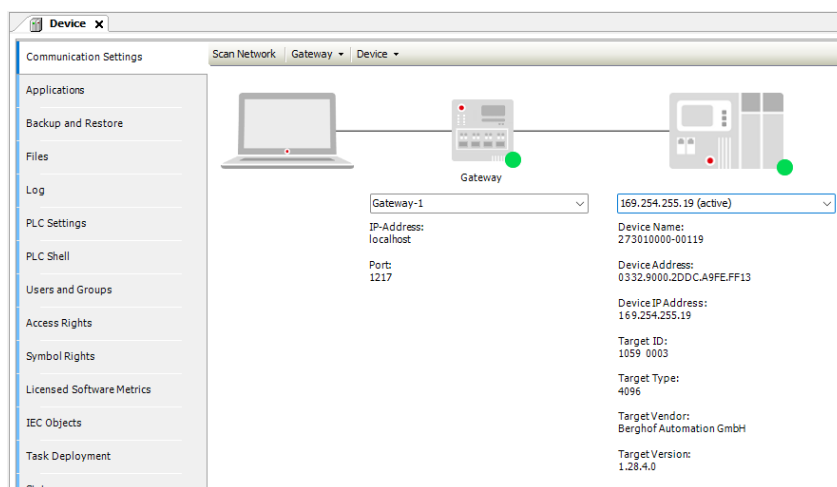
W tym punkcie zostanie opisany krok po kroku, proces tworzenia struktury drzewka projektowego, która konieczna jest do komunikacji Modbus TCP - docelowo powinna ona wyglądać następująco:



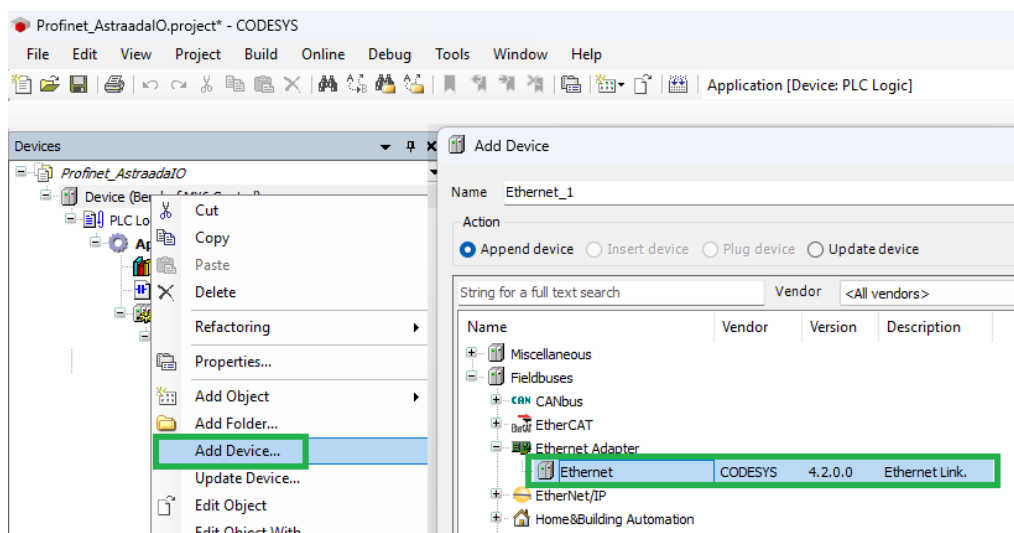
Należy pamiętać o zgodności podsieci adresów IP karty Ethernet oraz modułu Astraada IO.

Proces konfiguracji komunikacji Modbus TCP

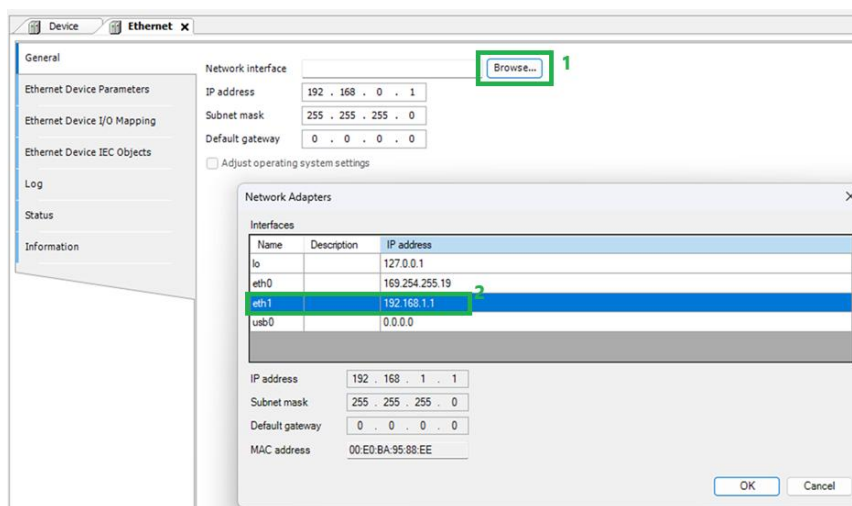
W docelowym projekcie, przechodzimy do „Device -> Communication Settings”, określamy ścieżkę komunikacji ze sterownikiem poprzez podanie jego adresu IP – powinniśmy zobaczyć zieloną kropkę przy ikonie sterownika:



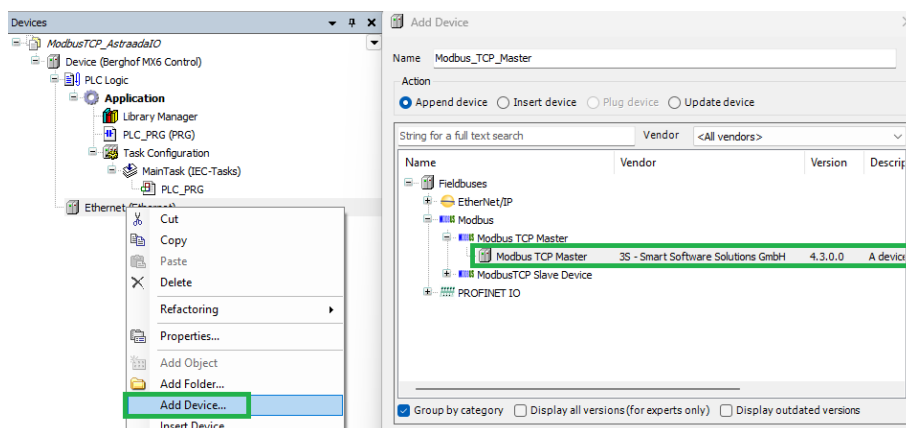
W drzewku projektowym klikamy prawym przyciskiem myszy na „Device” i wybieramy „Add Device...”. Z listy wybieramy „Fieldbuses -> Ethernet Adapter -> Ethernet”.



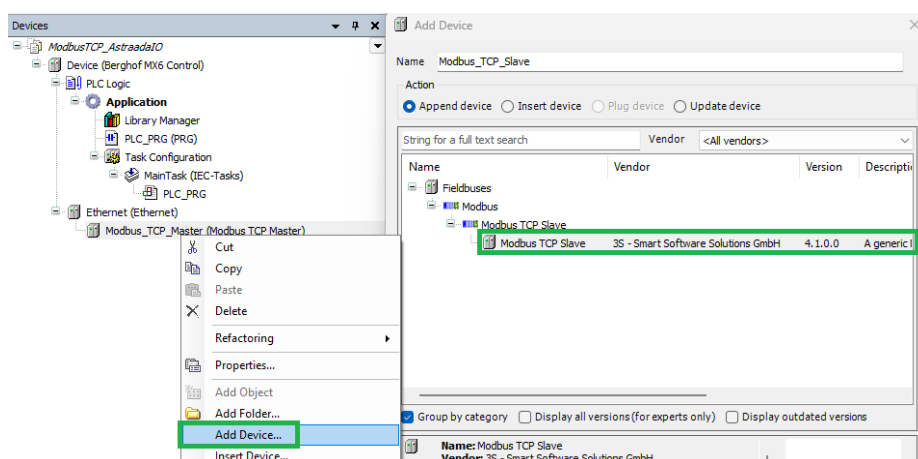
Przechodząc do właściwości urządzenia „Ethernet” w zakładce „General” wybieramy interfejs sieciowy do którego podłączony będzie moduł Astraada IO. Upewniamy się, że sterownik ma skonfigurowane docelowe adresy IP (zakładka „Network” w webserwerze sterownika).



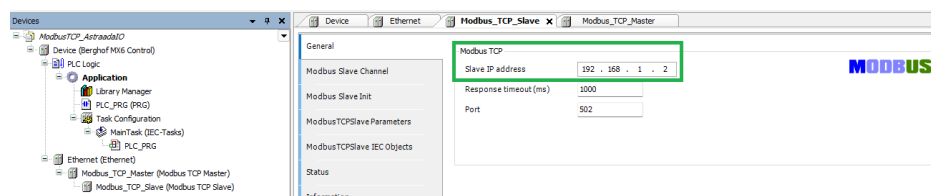
Klikamy prawym przyciskiem myszy na obiekt „Ethernet”, wybieramy „Add Device” i z listy wyszukujemy „Fieldbuses -> Modbus -> Modbus TCP Master -> Modbus TCP Master”.



W tym momencie, podobnie jak pozostałe urządzenia, dodajemy „Modbus TCP Slave”.



We właściwościach „Modbus TCP Slave”, zmieniamy automatycznie nadany adres IP na ten, który został skonfigurowany w module.



W tym miejscu kończymy konfigurację drzewka projektowego.

Struktura przesyłanych danych

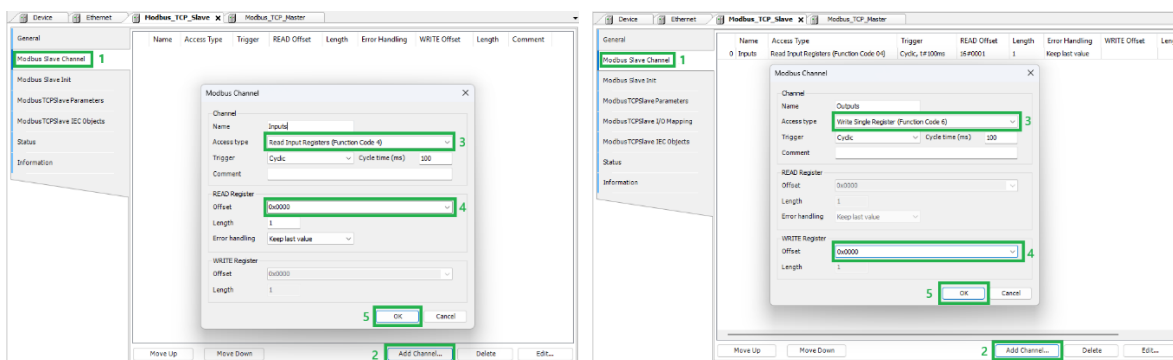
Wyspa sumarycznie udostępnia 3 bajty danych tj. stan wyjść, stan wejść oraz status modułu.

Modbus TCP – mapowanie przestrzeni wejść								
Adres (Discrete Input)	10008	10007	10006	10005	10004	10003	10002	10001
Adres (Input registers)	30001.7	30001.6	30001.5	30001.4	30001.3	30001.2	30001.1	30001.0
Stan wejść	Port 3 Pin 2	Port 3 Pin 4	Port 2 Pin 2	Port 2 Pin 4	Port 1 Pin 2	Port 1 Pin 4	Port 0 Pin 2	Port 0 Pin 4
Adres (Discrete Input)	10016	10015	10014	10013	10012	10011	10010	10009
Adres (Input registers)	30001.15	30001.14	30001.13	30001.12	30001.11	30001.10	30001.9	30001.8
Status Modułu				Zbyt wysokie napięcie US	Zbyt wysokie napięcie UA	Zbyt wysoka temp. pracy	Zbyt niskie napięcie US	Zbyt niskie napięcie UA

Modbus TCP – mapowanie przestrzeni wyjść								
Adres (Coil)	8	7	6	5	4	3	2	1
Adres (Holding registers)	40001.7	40001.6	40001.5	40001.4	40001.3	40001.2	40001.1	40001.0
Stan wyjść	Port 3 Pin 2	Port 3 Pin 4	Port 2 Pin 2	Port 2 Pin 4	Port 1 Pin 2	Port 1 Pin 4	Port 0 Pin 2	Port 0 Pin 4

Proszę zauważyć, że moduły AS70-R-MP-08DIO-P-M12 są wyposażone w wejścia adaptacyjne – tj. działają zawsze jako wejścia cyfrowe, natomiast w przypadku otrzymania rozkazu wystawienia wyjścia na danym pinie dodatkowo stają się wyjściem cyfrowym.

W celu konfiguracji wymiany danych przechodzimy do właściwości „Modbus TCP Slave”, wchodzimy do zakładki „Modbus Slave Channel” i wybieramy „Add Channel...”. Przykładowo, konfigurujemy jeden kanał odpytujący stan wejść oraz status i jeden kanał wysyłający zadany stan wyjść:



General								
Name	Access Type	Trigger	READ Offset	Length	Error Handling	WRITE Offset	Length	
0 Inputs	Read Input Registers (Function Code 04)	Cyclic, t#100ms	16#0000	1	Keep last value			
1 Outputs	Write Single Register (Function Code 06)	Cyclic, t#100ms				16#0000	1	

POTWIERDZENIE WYMIANY DANYCH I ICH INTERPRETACJA

We właściwościach wyspy „Modbus TCP Slave” przechodzimy do zakładki „ModbusTCP Slave I/O Mapping” i ustawiamy „Always update variables” na wartość „Enabled 2 (always in bus cycle task)” tak, aby móc w trybie online podglądać aktualne wartości zmiennych.

Po podłączeniu czujników w bajcie danych wejściowych widocznym jest, że zachodzi poprawna wymiana danych.

Variable	Mapping	Channel	Address	Type	Default Value	Current Value	Prepared Value
	Inputs	%DI0	ARRAY [0..3] OF WORD			Only subelements updated	
	Inputs[0]	%DI0	WORD		38		
	Bit0	%DI0.0	BOOL		FALSE		
	Bit1	%DI0.1	BOOL		TRUE		
	Bit2	%DI0.2	BOOL		TRUE		
	Bit3	%DI0.3	BOOL		FALSE		
	Bit4	%DI0.4	BOOL		FALSE		
	Bit5	%DI0.5	BOOL		TRUE		
	Bit6	%DI0.6	BOOL		FALSE		
	Bit7	%DI0.7	BOOL		FALSE		
	Bit8	%DI1.0	BOOL		FALSE		
	Bit9	%DI1.1	BOOL		FALSE		

Należy tutaj zauważyć istotną cechę wcześniej opisanych portów adaptacyjnych:

- wystawienie stanu wysokiego np. bitu 4 w kanale „Outputs” (tj. rozkazanie modułowi używania Pin 4 na Port 2 jako DO) zawsze powoduje ustawienie stanu wysokiego na korespondującym bicie kanale „Inputs”

Innymi słowy wszystkie bity ustawione na stan wysoki w kanale „Outputs” mają również stan wysoki w kanale „Inputs”, natomiast odwrotna sytuacja nie zawsze jest prawdziwa.

Outputs

Find	Filter	Show all	+ Add FB for IO Channel... Go to Instance				
Variable	Mapping	Channel	Address	Type	Default Value	Current Value	Prepared
		Standard Output	%Q0	ARRAY [0..3] OF BYTE		Only subelements updated	
		Standard Output[0]	%Q0	BYTE	20		
		Bit0	%Q0.0	BOOL	FALSE		
		Bit1	%Q0.1	BOOL	FALSE		
		Bit2	%Q0.2	BOOL	TRUE		
		Bit3	%Q0.3	BOOL	FALSE		
		Bit4	%Q0.4	BOOL	TRUE		
		Bit5	%Q0.5	BOOL	FALSE		
		Bit6	%Q0.6	BOOL	FALSE		
		Bit7	%Q0.7	BOOL	FALSE		
		Outputs CS	%Q0	Enumeration of BYTE	GOOD		
Reset Mapping Always update variables Enabled 2 (always in bus cycle task)							

+ Create new variable

+ Map to existing variable

Inputs

Find	Filter	Show all	+ Add FB for IO Channel... Go to Instance				
Variable	Mapping	Channel	Address	Type	Default Value	Current Value	Prepared
		Standard Input	%I0	ARRAY [0..3] OF BYTE		Only subelements updated	
		Standard Input[0]	%I0	BYTE	22		
		Bit0	%I0.0	BOOL	FALSE		
		Bit1	%I0.1	BOOL	FALSE		
		Bit2	%I0.2	BOOL	FALSE		
		Bit3	%I0.3	BOOL	FALSE		
		Bit4	%I0.4	BOOL	TRUE		
		Bit5	%I0.5	BOOL	FALSE		
		Bit6	%I0.6	BOOL	FALSE		
		Bit7	%I0.7	BOOL	FALSE		
		Inputs PS	%I0	Enumeration of BYTE	GOOD		
Reset Mapping Always update variables Enabled 2 (always in bus cycle task)							

+ Create new variable

+ Map to existing variable

Takie zachowanie spowodowane jest faktem, że pojawienie się napięcia wyjściowego na pinie jest równocześnie odczytywane przez czujnik wejścia.