

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.

ASTOR

KONFIGURACJA MODUŁU ASTRAADA IO

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

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 <u>stronie wsparcia</u> <u>Astor</u>.

- 1. Uruchom program Astraada IO Configuration Tool,
- 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.



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.



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

ASTOR

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:

Device X				
Communication Settings	Scan Network Gateway -	Device 🔹		
Applications				
Backup and Restore			• • • • • • • • • • • • • • • • • • •	
Files				•
Log			Gateway	
PLC Settings		Gateway-1		169.254.255.19 (active)
PI C Shell		localhost		273010000-00119
		Port: 1217		Device Address: 0332.9000.2DDC.A9FE.FF13
Users and Groups				Device IP Address:
Access Rights				Target ID:
Symbol Rights				1059 0003
Licensed Software Metrics				Target Type: 4096
IEC Objects				Target Vendor: Berghof Automation GmbH
Task Deployment				Target Version: 1.28.4.0
Status				



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

💗 Profinet_Astraada	lO.pr	oject* - CODESYS											
File Edit View	P	roject Build (Online Debug	То	ols V	/indow Help							
1 🚔 🖬 🖓 🖿	2 0	3 B B X	🗛 😘 🐴 🌿		(%) ⁻	1 1 🐴 🛱 🏪 • 🗗 i	🛗 Applic	ation [[Device: PLC	Logic]			
Þ.													
Devices			▼ 7	X (🚹 Add	Device							
Profinet_Astra	adaIO	,		•									
🖹 💮 Device (Be	0	C		-	Name	Ethernet_1							
🖹 🗐 PLC Lo	ð	Cut			Action								
🖹 🔘 AI		Сору				oend device 🔘 Insert devi	ice 🔿 Pluc	device	O Update	e device			
- É	Ē	Paste											
	\times	Delete			String	for a full text search		Ver	ndor <all< td=""><td>vendors></td><td></td></all<>	vendors>			
		Defendenting	•	1	Nam	2	Ven	dor	Version	Description			
		Refactoring											
	Ē.	Properties		1	P - 1	Miscellaneous							
	i i i i i i i i i i i i i i i i i i i	Properties			• • •	Miscellaneous Fieldbuses							
		Properties Add Object	•			Miscellaneous Fieldbuses CANbus							
		Properties Add Object Add Folder	•			Miscellaneous Fieldbuses CAN CANbus Brott EtherCAT							
		Properties Add Object Add Folder Add Device	•	-		Miscellaneous Fieldbuses Budt CANbus Budt EtherCAT					_		
		Properties Add Object Add Folder Add Device Update Device	•			Miscellaneous Fieldbuses CAN CANbus Brot EtherCAT Ethernet Adapter Ethernet	COD	ESYS	4.2.0.0	Ethernet Link.			
		Properties Add Object Add Folder Add Device Update Device Edit Object	•			Miscellaneous Fieldbuses CAN CANbus But EtherCAT Ethernet Adapter Ethernet EtherNet/IP	COD	ESYS	4.2.0.0	Ethernet Link.			

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).

General	Network interface			Browse 1	
Ethernet Device Parameters	IP address	192 . 168	. 0 . 1		
Ethernet Device I/O Mapping	Subnet mask	255 . 255	. 255 . 0		
Ethernet Device IEC Objects	Default gateway	0.0	. 0 . 0		
Log	Adjust operatin	g system setti	igs		
Status	Interfaces	adapters			~
Information	Name	Description	IP address		
	lo	_	127.0.0.1		
	eth0		169.254.255.19	2	
	eth1		192.168.1.1		6
	usb0		0.0.0.0		
	IP addres	s [1	92,168,1,1]	
	Subnet m	ask 2	55 . 255 . 255 . 0		
	Default ga	steway	0.0.0.0		
	MAC add	ress <u>0</u>	E0:BA:95:88:EE		
					OK Cancel

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



Devices	→ ‡ X	Add D	levice				×
Monadar I/2_Astrahatal Manager Manager PLC Logic Application PLC Reg (PRG) PLC PRG (PRG) WanTask (Efc-Tasks) Manager Manager		Name M Action Appe String for Name	lodbus_TCP_Master Ind device ○ Insert device ○ ra full text search Fieldbuses ← EtherNet/IP Modbus	Plug device O U Vendor Vendor	Ipdate device <all vendors=""></all>	Version	
Image: Copy Image: Copy <t< td=""><td>,</td><td>€} </td><td>Mi Modbus TCP Master Modbus TCP Master Mit Modbus TCP Salve Termine Mit Modbus TCP Salve Device PROFINET IO</td><td>3S - Smart Soft</td><td>ware Solutions GmbH</td><td>4.3.0.0</td><td>A device</td></t<>	,	€} 	Mi Modbus TCP Master Modbus TCP Master Mit Modbus TCP Salve Termine Mit Modbus TCP Salve Device PROFINET IO	3S - Smart Soft	ware Solutions GmbH	4.3.0.0	A device

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

Devices 👻 🕈 🗙	If Add Device X
ModbusTCP_Astraadat0 ModbusTCP_Astraadat0 ModbusTCP_Astraadat0 ModbusTCP_Astraadat0 ModbusTCP_Astraadat0 Mit Device (Berghof MKS Control) Mit Device (Berghof MKS Cont	Name Modbus_TCP_Slave Action O Append device String for a full text search Vendor Vanne Vendor Version Descriptiv Image: Instructure Vendor
Modbus_TCP_Master Modbus_TCP_Master % Cut % Cut % Paste % Delete Refactoring Refactoring % Properties	Init Modbus TCP Slave Modbus TCP Slave 3S - Smart Software Solutions GmbH 4.1.0.0 A generic 1
Add Object Add Folder Add Device Inset Device	Group by category Display all versions (for experts only) Display outdated versions Name: Modbus TCP Slave Vendor: 33 - Smart Software Solutions GmbH

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

Devices 👻	4 X	Device 🔐 Ethernet	Modbus_TCP_Slave 🗙 📑	Modbus_TCP_Master	
Modbus70P_Astraada10 Modbus70P_Astraa	•	General Modbus Slave Channel	- Modbus TCP Slave IP address	192 . 168 . 1 . 2	MODBUS
Contraction Definition Definition Definition PLC PRG (PRG)	Modbus Slave Init	Response timeout (ms)	1000		
🖻 🗱 Task Configuration		ModbusTCPSlave Parameters	Port	502	
= ∰ MainTask (IEC-Tasks) - ∰ PLC_PRG =-∭ Ethernet (Ethernet)		ModbusTCPSlave IEC Objects			
Modbus_TCP_Master (Modbus TCP Master) Modbus TCP State Modbus TCP State		Status			
I House_TCP_state (House TCP state)		Information			

W tym miejscu kończymy konfigurację drzewka projektowego.

ASTOR

Struktura przesyłanych danych

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						

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

	Modbus TCP – mapowanie przestrzeni wyjść													
Adres (Coil)	8	7	6	6 5		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:



Device 🔐 Ethemet	Hodbus_TCP_Slave X Modus_TCP_Nater	Cevice Guthernet Guth
General	Name Access Type Trigger READ Offset Length Error Handling WRITE Offset Length Comment	General Neme Access Type Trigger READ Offset Length Enor Handling WRITE Offset Lengt 0 Inputs Read Input Repeters (Function Code 04) Cyclic, tristOms 36:00001 1 Repeters value
Madaw Sava Canwal Madaw Sava Sat Madaw (Silare Parates Madaw (Silare Raines Raine Difernation	Mothes Diamed X Oursel Accessity and type Argeners Puncters Code dt Accessity and type Argeners Puncters Code d	Implicit Stars Channel 0 Percel Read To Stars Channel Very batt rule Models Stars Date Models Channel Very batt rule Very batt rule
	Move Up Move Down 2 Add Channel Delete Edt	Prove up in over power 2 Add chambers particle 2015

	Device x M Ethernet Modbus_TCP_Slave x M Kodbus_TCP_Master												
ſ		_									1		
	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					
	Modbus Slave Channel		Outputs	Write Single Register (Function Code 06)	Cyclic, t#100ms				16#0000	1			
											Ш		



POTWIERDZENIE WYMIANY DANYCH I ICH INTERPRETACJA

We właściwościach wyspy "Modbus TCP Slave" przechodzimy do zakładki "ModbusTCPSlave 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.

Find		Filter Show a		•	Add FB for IO 0	Channel * Go to Instance	
Variable	Mapping	Channel	Address	Туре	Default Value	Current Value	Prepared Value
8- %		Inputs	%IW0	ARRAY [00] OF WORD		Only subelements updated	
🖹 - 🎭		Inputs[0]	%IW0	WORD		38	
*•		Bit0	%IX0.0	BOOL		FALSE	
- 🍫		Bit1	%EX0.1	BOOL		TRUE	
		Bit2	%IX0.2	BOOL		TRUE	
**		Bit3	%IX0.3	BOOL		FALSE	
- *9		Bit4	%IX0.4	BOOL		FALSE	
- **		Bit5	%IX0.5	BOOL		TRUE	
🍫		Bit6	%IX0.6	BOOL		FALSE	
🍫		Bit7	%IX0.7	BOOL		FALSE	
🍫		Bit8	%JX1.0	BOOL		FALSE	
- *9		Bit9	96DX1.1	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.



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