

POŁĄCZENIE STEROWNIKÓW ASTRAADA ONE MIĘDZY SOBĄ Z WYKORZYSTANIEM PROTOKOŁU UDP

Sterowniki Astraada One wymieniają między sobą dane po UDP

Wstęp

Celem informatora jest konfiguracja i przygotowanie sterowników Astraada One do wzajemnej współpracy i wymiany danych pomiędzy sobą.

Sterowniki Astraada One mają w standardzie protokół EtherCAT Master. Jednak w tym protokole nie jesteśmy w stanie wymieniać danych pomiędzy sterownikami (jednostkami CPU).

Dlatego poniżej zostanie przedstawiona konfiguracja umożliwiająca dane pomiędzy sterownikami Astraada One.



Protokół UDP

UDP (ang. *User Datagram Protocol* – protokół pakietów użytkownika) – UDP stosowany jest w warstwie transportowej modelu OSI. Nie gwarantuje dostarczenia datagramu.

Jest to protokół bezpołączeniowy, więc nie ma narzutu na nawiązywanie połączenia i śledzenie sesji (w przeciwieństwie do TCP). Nie ma też mechanizmów kontroli przepływu i retransmisji. Korzyścią płynącą z takiego uproszczenia budowy jest większa szybkość transmisji danych i brak dodatkowych zadań, którymi musi zajmować się host obsługujący się tym protokołem. Z tych względów UDP jest często używany, gdzie dane muszą być przesyłane możliwie szybko, a poprawianiem błędów zajmują się inne warstwy modelu OSI.

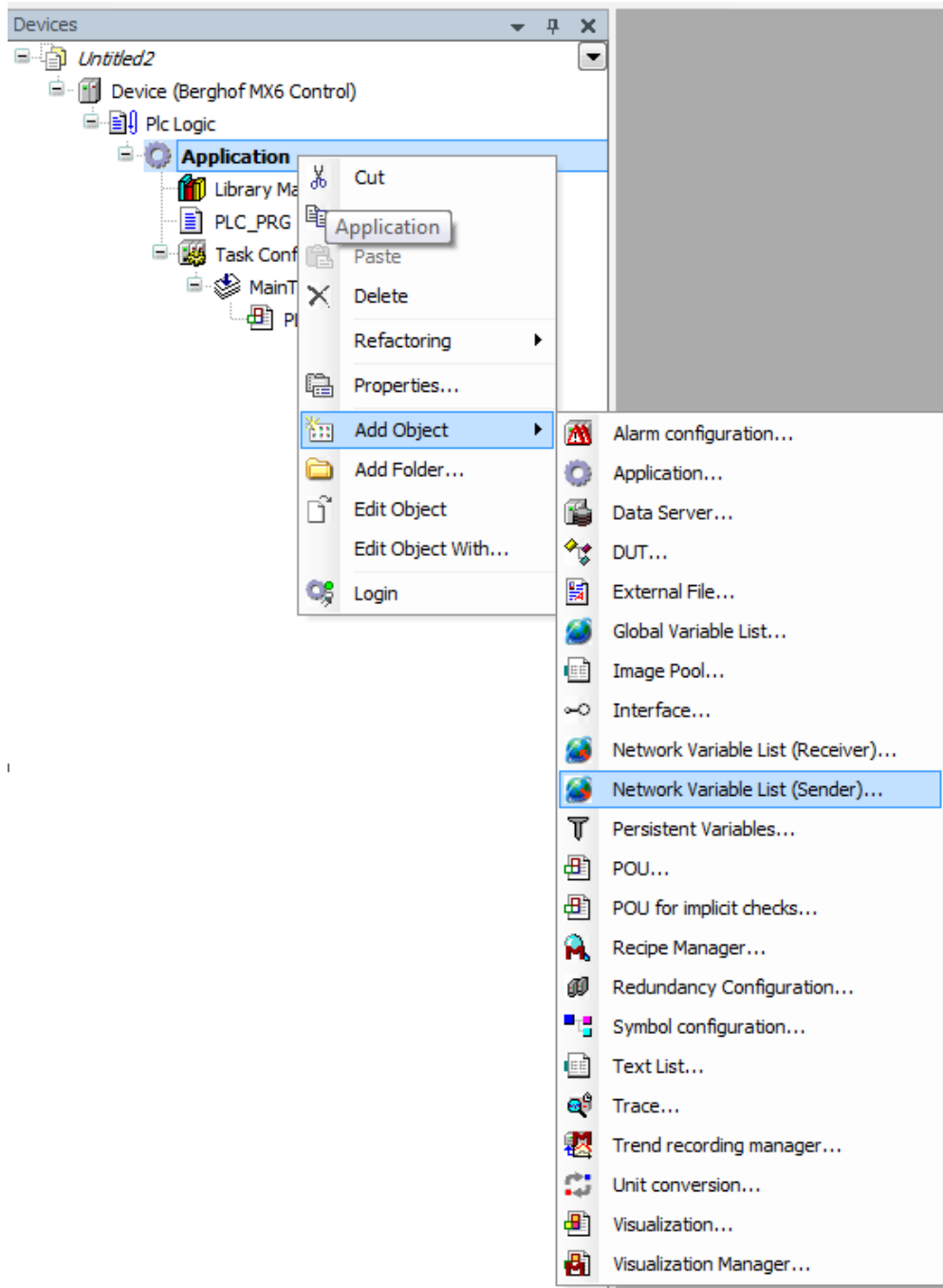
Konfiguracja

Po założeniu standardowego projektu w drzewku konfiguracyjnym należy dodać Network Variable List Sender oraz Receiver.

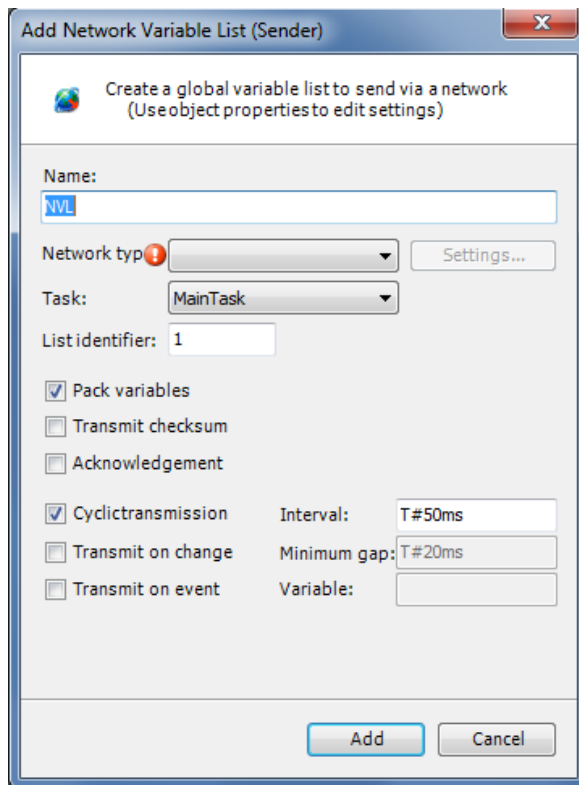
Najpierw skonfigurujemy zmienne do wysyłania:

1. Network Variable List (Sender):

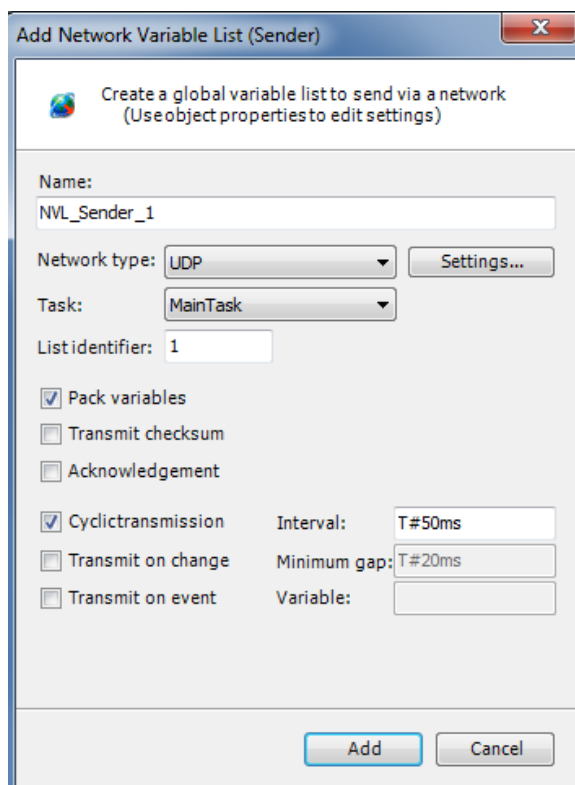
W tym celu klikamy PPM na Application -> Add Object -> Network Variable List (Sender):



Po dodaniu Network Variable List Sender, pojawi się okno konfiguracyjne:



Gdzie, wpisujemy nazwę, definiujemy typ połączenia i jego ustawienia:



Add Network Variable List (Sender)

Create a global variable list to send via a network
 (Use object properties to edit settings)

Name: NVL_Sender_1

Network type: UDP Settings...

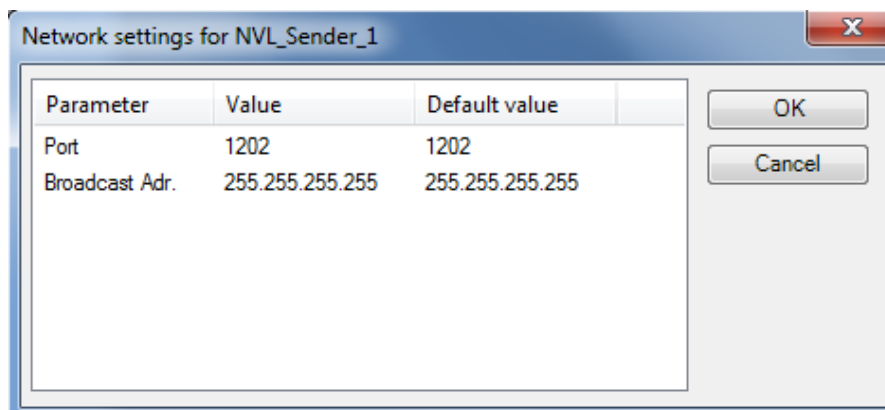
Task: MainTask

List identifier: 1

Pack variables
 Transmit checksum
 Acknowledgement
 Cyclictransmission Interval: T#50ms
 Transmit on change Minimum gap: T#20ms
 Transmit on event Variable:

Add Cancel

Klikamy w Settings:

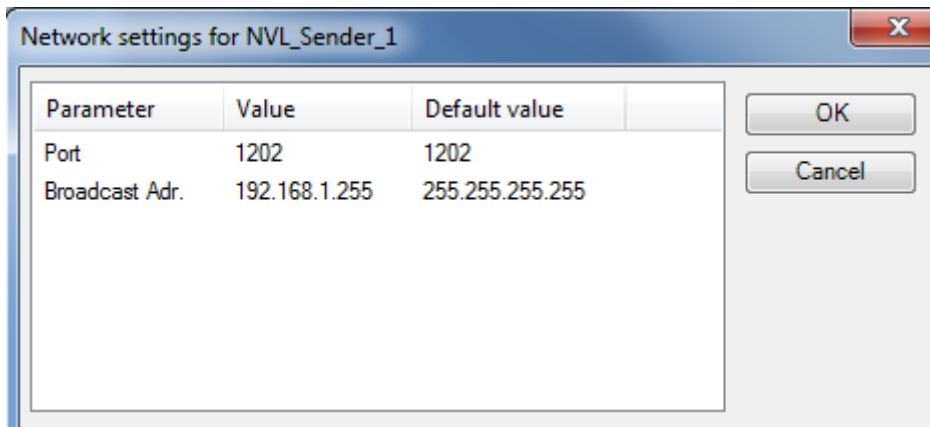


Network settings for NVL_Sender_1

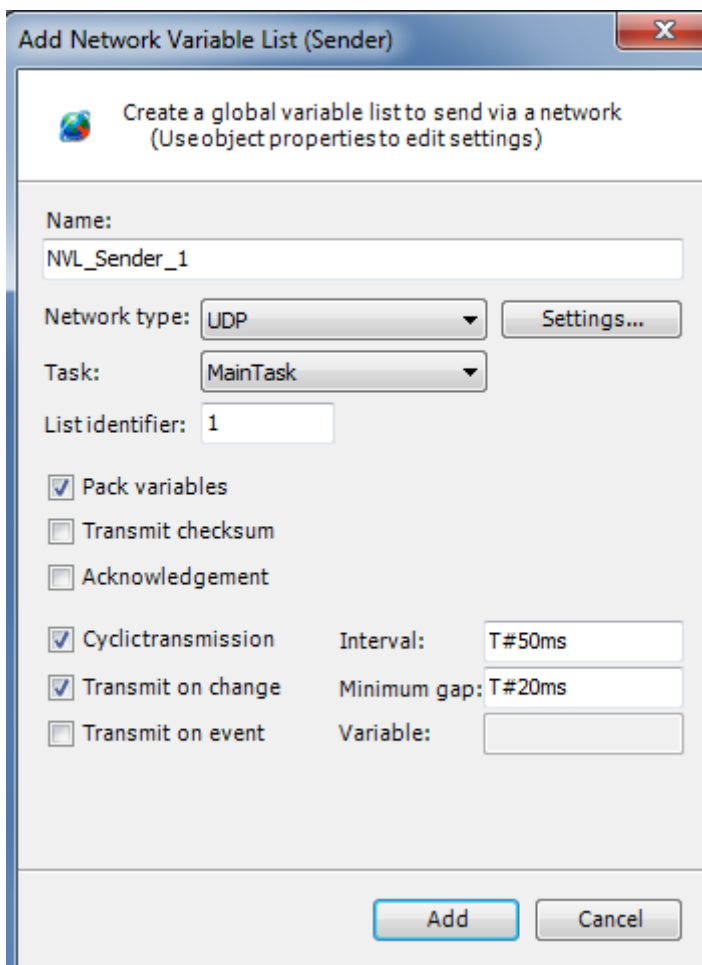
Parameter	Value	Default value
Port	1202	1202
Broadcast Adr.	255.255.255.255	255.255.255.255

OK
 Cancel

Tutaj ustawiamy port połączenia oraz możemy ograniczyć urządzenia do podłączenia po zmniejszeniu puli adresów IP.



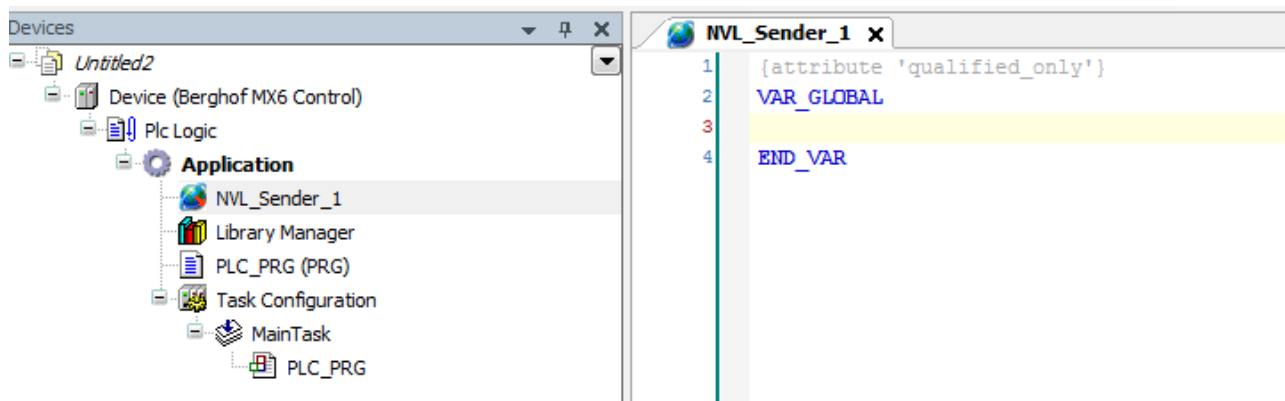
W tym przypadku została zmniejszona pula adresów IP do nawiązania połączenia i tylko urządzenia z puli 192.168.1.x będą mogły odbierać wysyłane dane, z tego urządzenia.



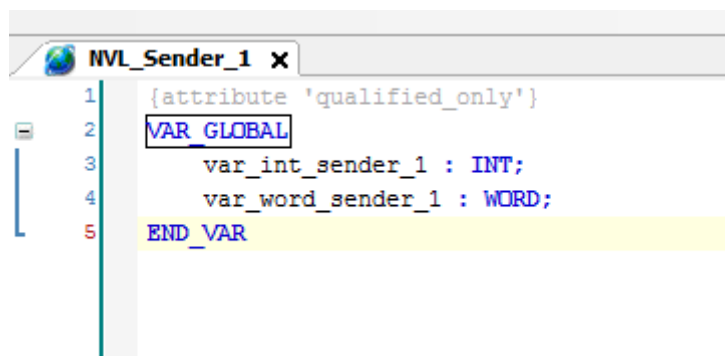
Bardzo istotne jest skonfigurowanie numeru identyfikacyjnego. Jeżeli będziemy korzystać z większej ilości Network Variable List (Sender), gdzie ustawimy ten sam port np. 1202, należy każdą „paczkę” wysyłanych zmiennych ustawić z innym identyfikatorem.

Inne opcje są opcjonalne, powyżej ustawiono np. transmisję danych po zmianie wartości zmiennych.

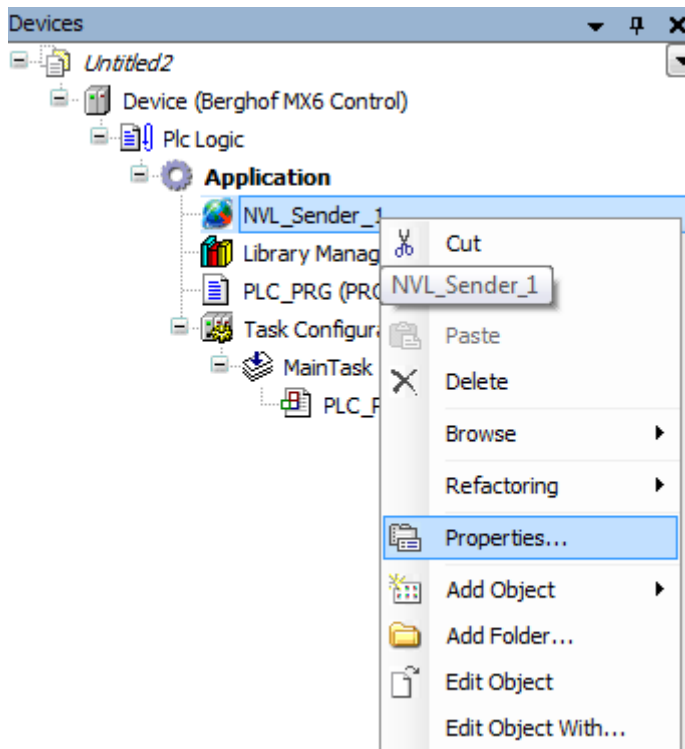
Potwierdzamy wybór klikając na Add:



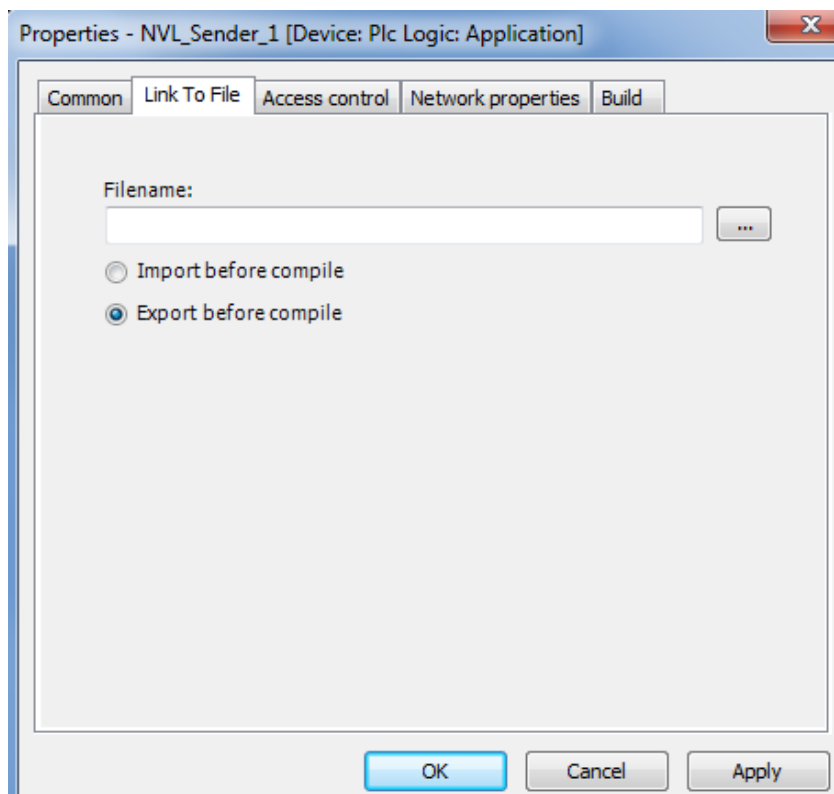
W oknie NVL_Sender_1 definiujemy zmienne, które będziemy wysyłać, przykładowo:



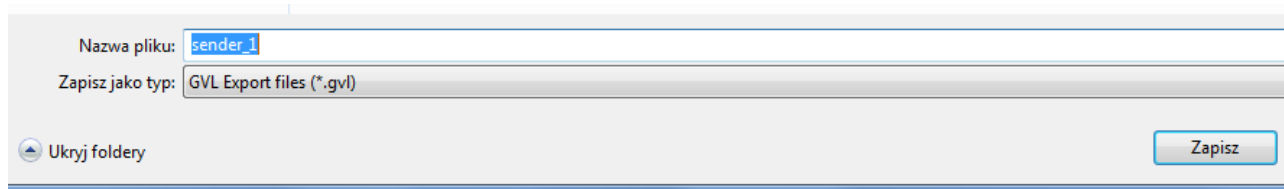
Następnie klikamy PPM na NVL_Sender_1 i wybieramy Properties:



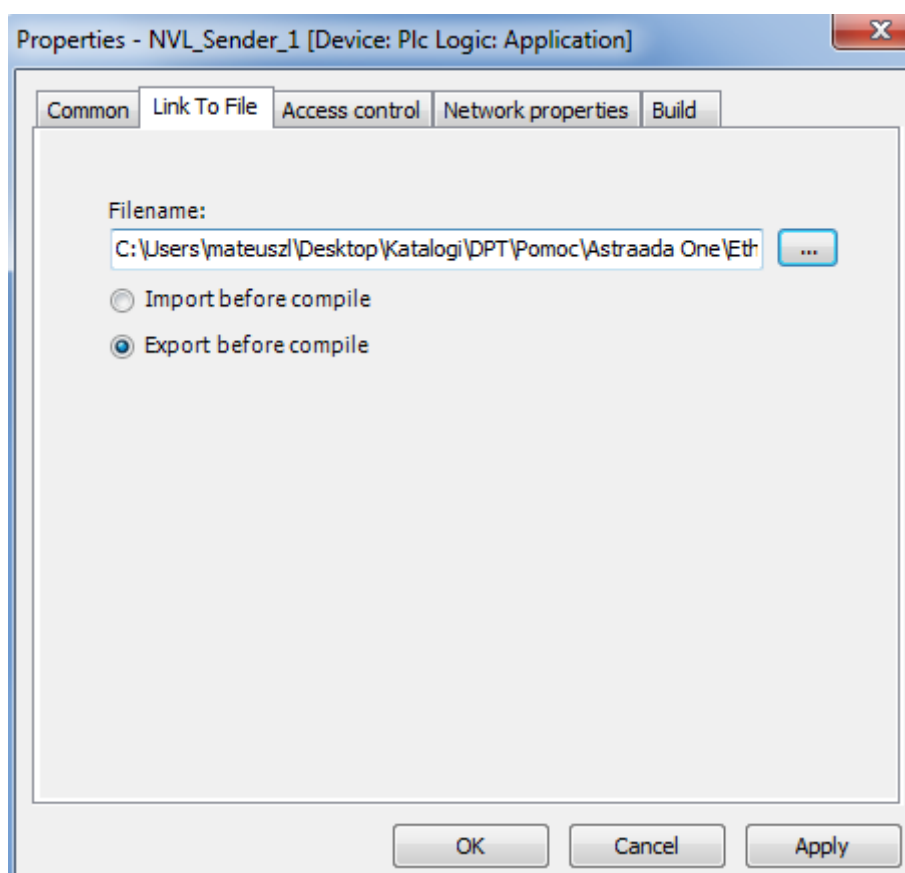
W otwartym oknie przechodzimy do zakładki Link to File:



W polu powyżej podajemy, plik, w którym będą zapisane nasze zmienne, najlepiej kliknąć w obiekt ... :



Zapisujemy:



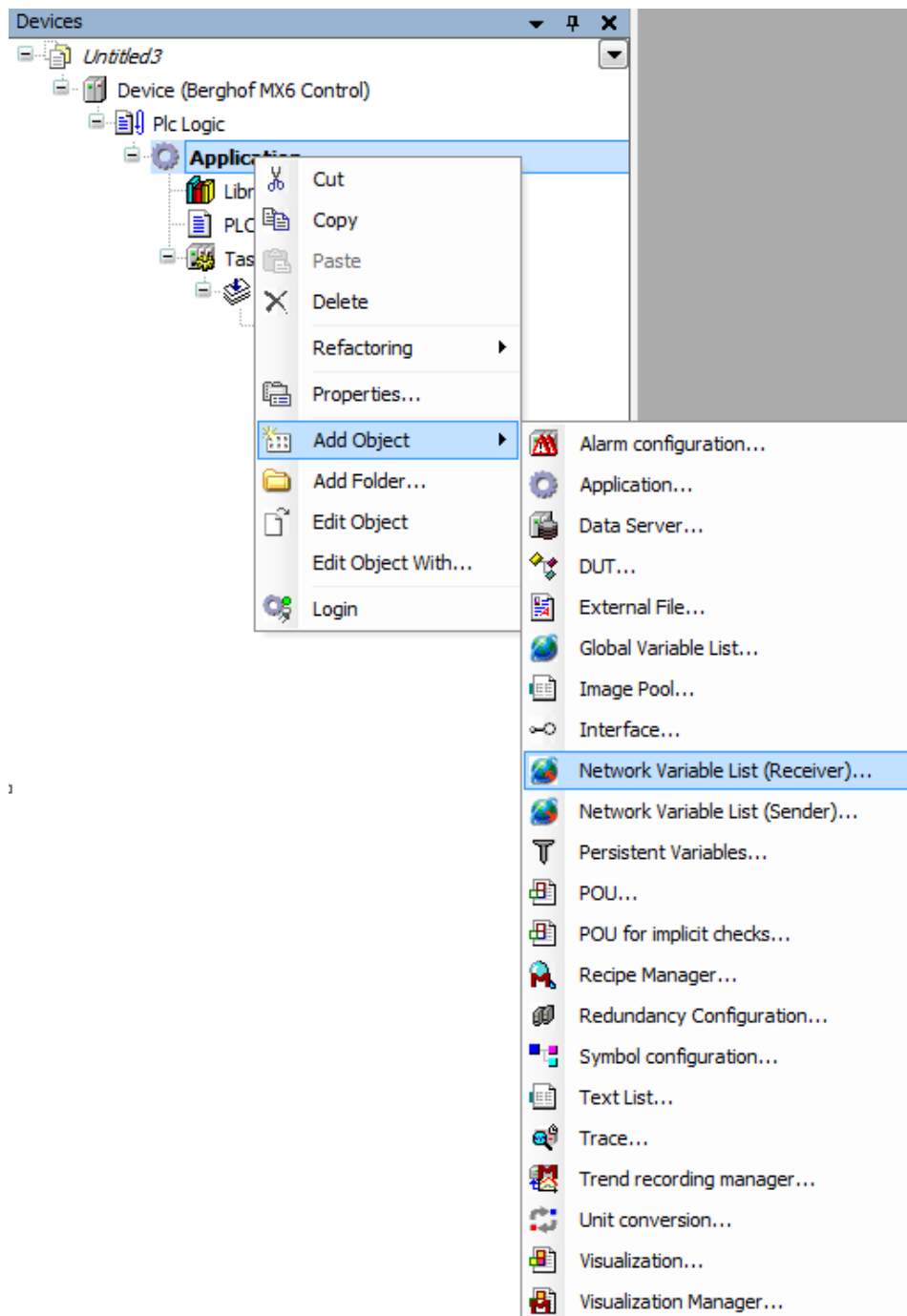
Potwierdzamy wybór i wykonujemy kompilację programu: Build -> Build lub wciskamy skrót klawiszowy F11.

Po wykonaniu kompilacji zostanie stworzony plik, do którego ścieżkę podaliśmy wyżej.

Wgrywamy aplikację do sterownika. Teraz możemy przejść do konfiguracji odbierania danych na drugim sterowniku, czyli Network Variable List Receiver.

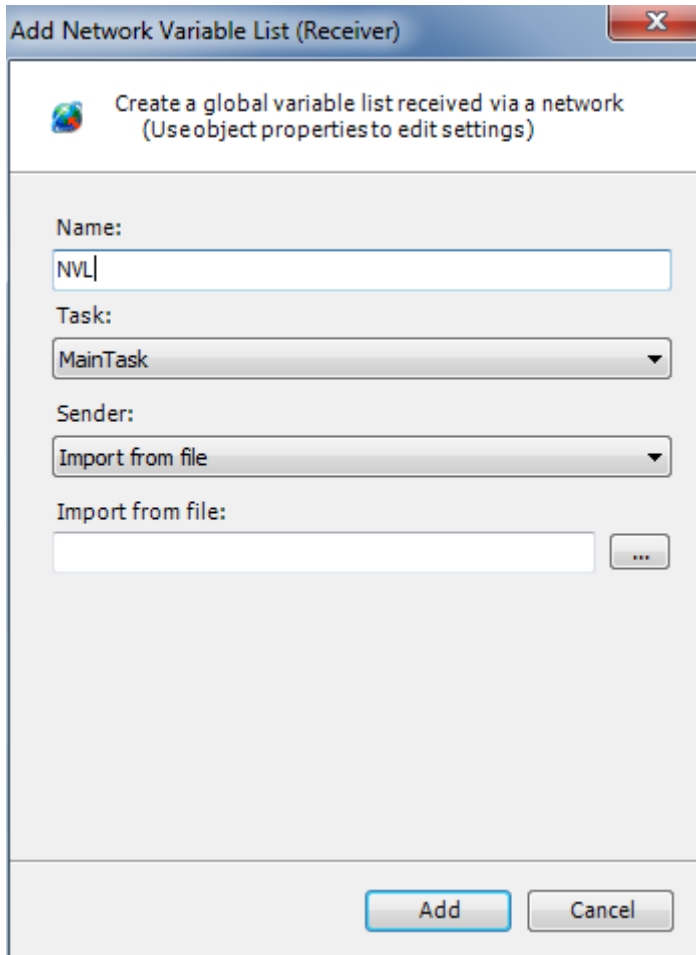
2. Network Variable List (Receiver)

Klikamy PPM na Application, wybieramy Add Object -> Network Variable List (Receiver):



1

Pojawi się następujące okno:



Add Network Variable List (Receiver)

Create a global variable list received via a network
(Use object properties to edit settings)

Name:

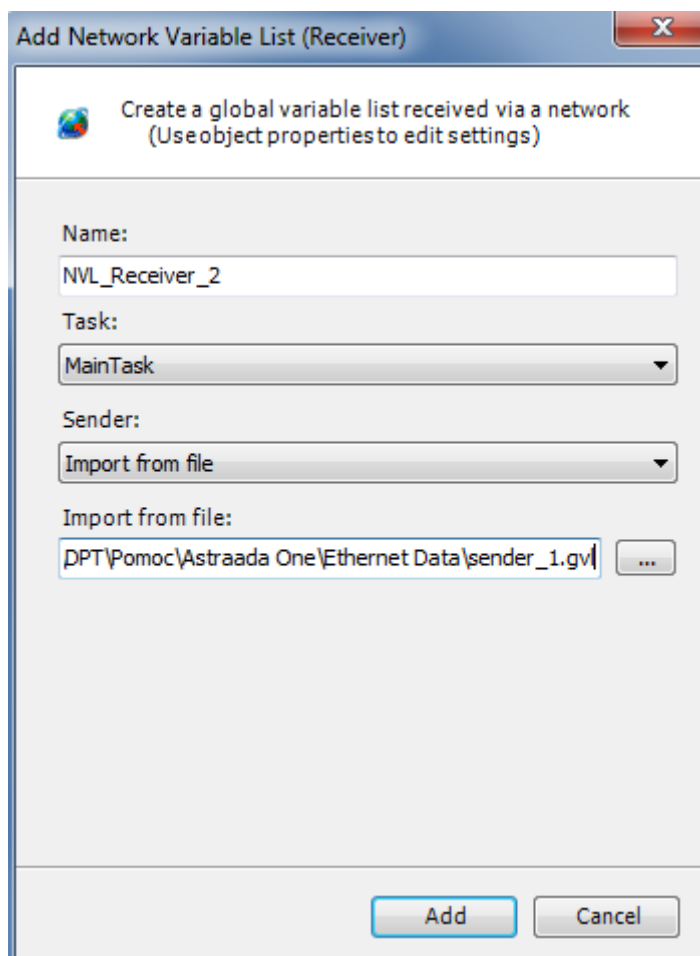
Task:

Sender:

Import from file:
 ...

Add Cancel

Gdzie, podajemy nazwę, Task, w którym odbiór danych zostanie wykonany oraz podajemy plik, pochodzący od danych wysyłanych:





Po kliknięciu w Add, zobaczymy okno, gdzie będziemy mieli już informację o zmiennych, które będziemy odbierać:



```

1 //This gobal variable list is received via the network.
2 //Sender: Imported from file 'C:\Users\mateusz1\Desktop\Katalogi\DPT\Pomoc\Astraada One\Ethernet Data\sender_1.gvl'
3 //Protocol: UDP
4
5 {attribute 'qualified_only'}
6 VAR_GLOBAL
7     var_int_sender_1 : INT;
8     var_word_sender_1 : WORD;
9 END_VAR
    
```



Program, wgrujemy do drugiego sterownika.



Działanie programów:

Watch 1					
Expression	Application	Type	Value	Prepared value	Executionpoint
 NVL_Sender_1.var_int_sender_1	Device.Application	INT	0		Cyclic Monitoring
 NVL_Sender_1.var_word_sender_1	Device.Application	WORD	0		Cyclic Monitoring

Watch 1					
Expression	Application	Type	Value	Prepared value	Executionpoint
 NVL_Receiver_2.var_int_sender_1	Device.Application	INT	0		Cyclic Monitoring
 NVL_Receiver_2.var_word_sender_1	Device.Application	WORD	0		Cyclic Monitoring

Przygotujemy wartości do wysłania, odpowiednio np. 555 oraz 777:

Watch 1					
Expression	Application	Type	Value	Prepared value	Executionpoint
 NVL_Sender_1.var_int_sender_1	Device.Application	INT	555		Cyclic Monitoring
 NVL_Sender_1.var_word_sender_1	Device.Application	WORD	777		Cyclic Monitoring

Watch 1					
Expression	Application	Type	Value	Prepared value	Executionpoint
 NVL_Receiver_2.var_int_sender_1	Device.Application	INT	555		Cyclic Monitoring
 NVL_Receiver_2.var_word_sender_1	Device.Application	WORD	777		Cyclic Monitoring

Istnieje możliwość dodawania i łączenia wielu sterowników oraz tworzenia wielu Network List Variable Sender oraz Receiver.