

Konfiguracja komunikacji CANopen pomiędzy sterownikami: Astraada One (Master) i PacDrive LMC (Slave)

Informator techniczny PacDrive 3

Spis treści

Wstęp	1
Elementy wykorzystywane w poniższej instrukcji.....	1
Konfiguracja sterownika PacDrive LMC jako CANopen Slave	2
Budowanie struktury programu	2
Konfigurowanie modułów i urządzeń.....	7
Zapisywanie konfiguracji do pliku EDS	11
Wgrywanie programu do kontrolera.....	12
Mapowanie IO	14
Konfiguracja sterownika Astraada One jako CANopen Master w środowisku CODESYS.....	16
Budowanie struktury programu	16
Konfigurowanie modułów i urządzeń.....	22
Mapowanie IO	23
Wgrywanie programu do kontrolera.....	24
Monitorowanie wymiany danych w komunikacji CANopen	26

WSTĘP

Poniższy informator przedstawia, w jaki sposób skonfigurować sterowniki PacDrive 3 serii LMC Pro jako CANopen Slave oraz Astraada One jako CANopen Master, w celu ustanowienia komunikacji po protokole CANopen. W informatorze znajduje się osobne omówienie konfiguracji CANopen dla obu sterowników.

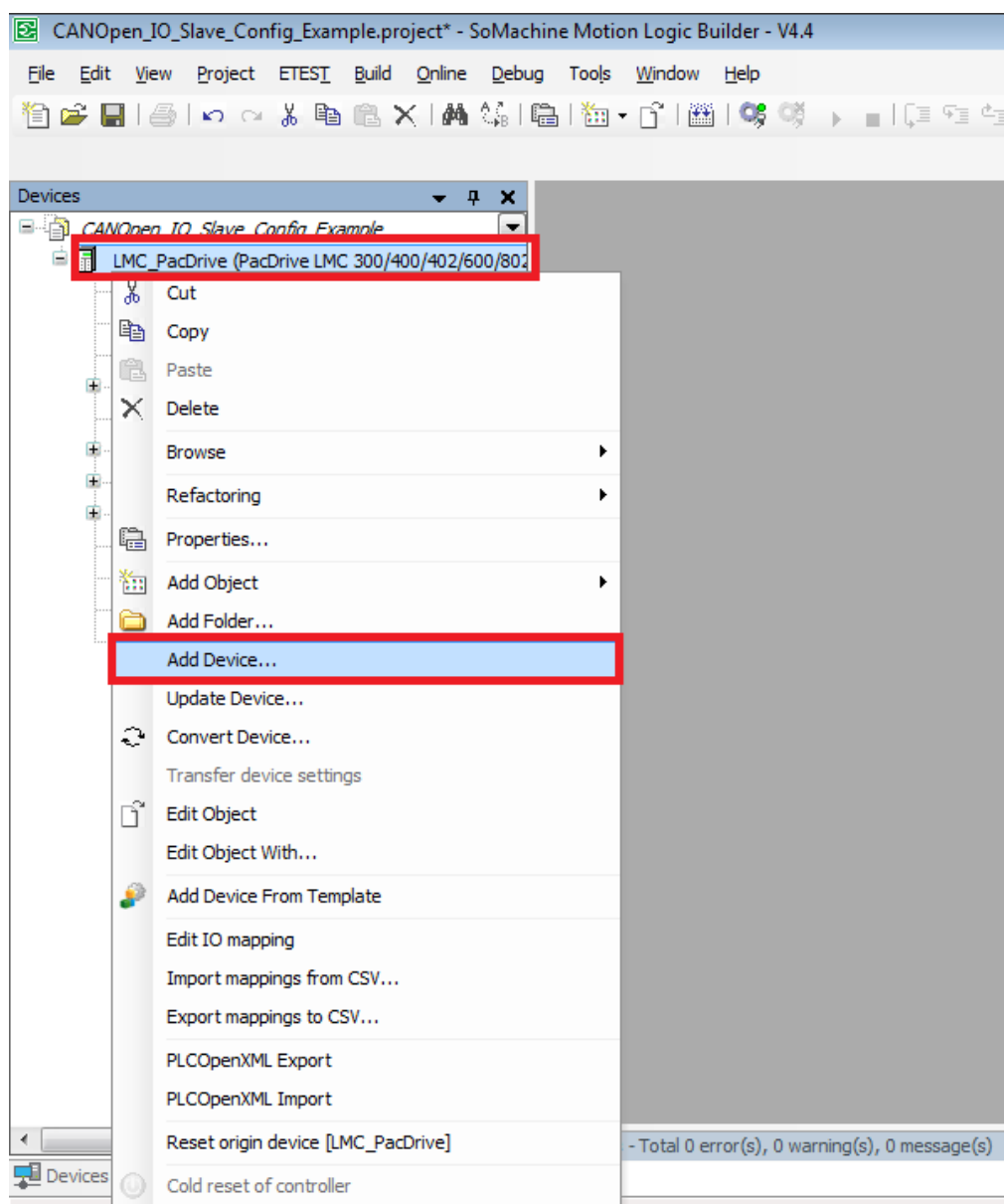
ELEMENTY WYKORZYSTYWANE W PONIŻSZEJ INSTRUKCJI

- Sterownik ruchu PacDrive 3 - LMC 300C
- Astraada One DC2004W

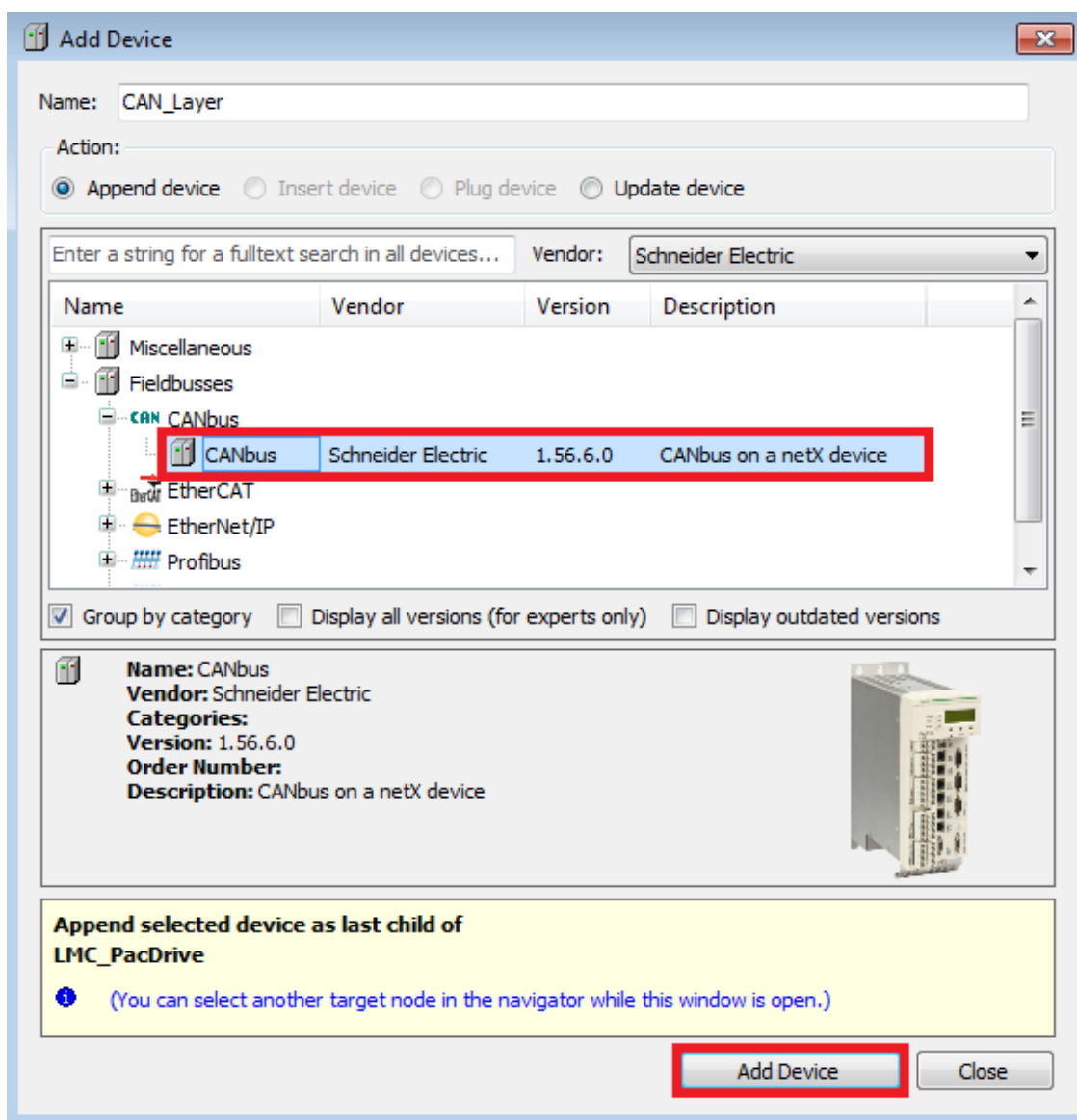
Konfiguracja sterownika PacDrive LMC jako CANopen Slave

BUDOWANIE STRUKTURY PROGRAMU

Pierwszym krokiem w konfiguracji protokołu CANopen jest zbudowanie odpowiedniej struktury komunikujących się urządzeń i danych, które między sobą wymieniają. Ustawiamy sterownik PacDrive 3 LMC 300C jako CANopen Slave. Aby umożliwić komunikację po wyżej wymienionym protokole, należy dodać moduł obsługujący protokoły CAN. W tym celu klikamy prawym przyciskiem myszy na urządzeniu *LMC_PACDrive* na liście urządzeń, a następnie wybieramy opcję *Add Device...*



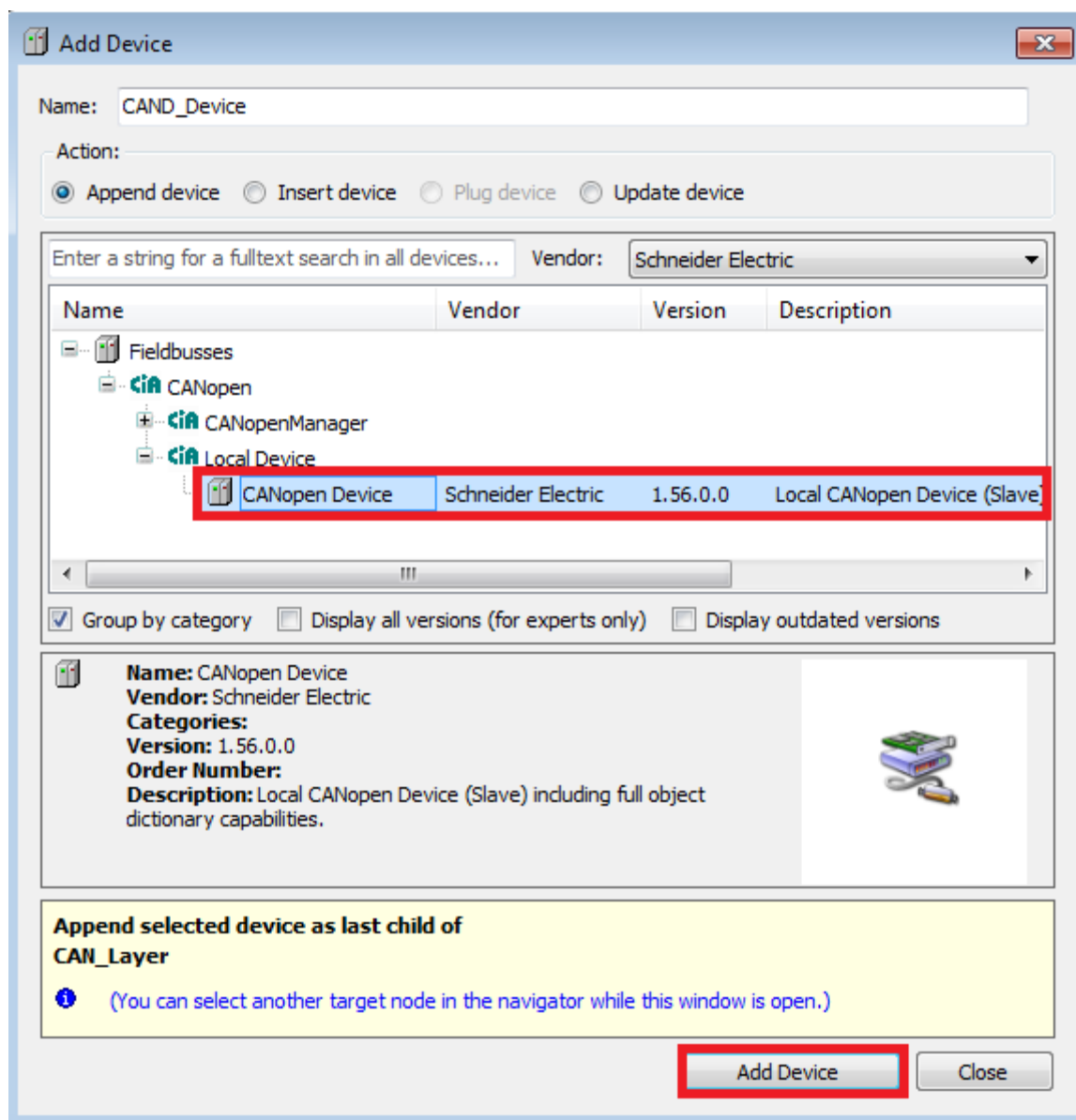
Pojawi się nowe okno, w którym rozwijamy listę *Fieldbusses*>*CANbus*, a następnie wybieramy urządzenie CANbus. Dodajemy je do struktury programu zatwierdzając wybór przyciskiem *Add device*.



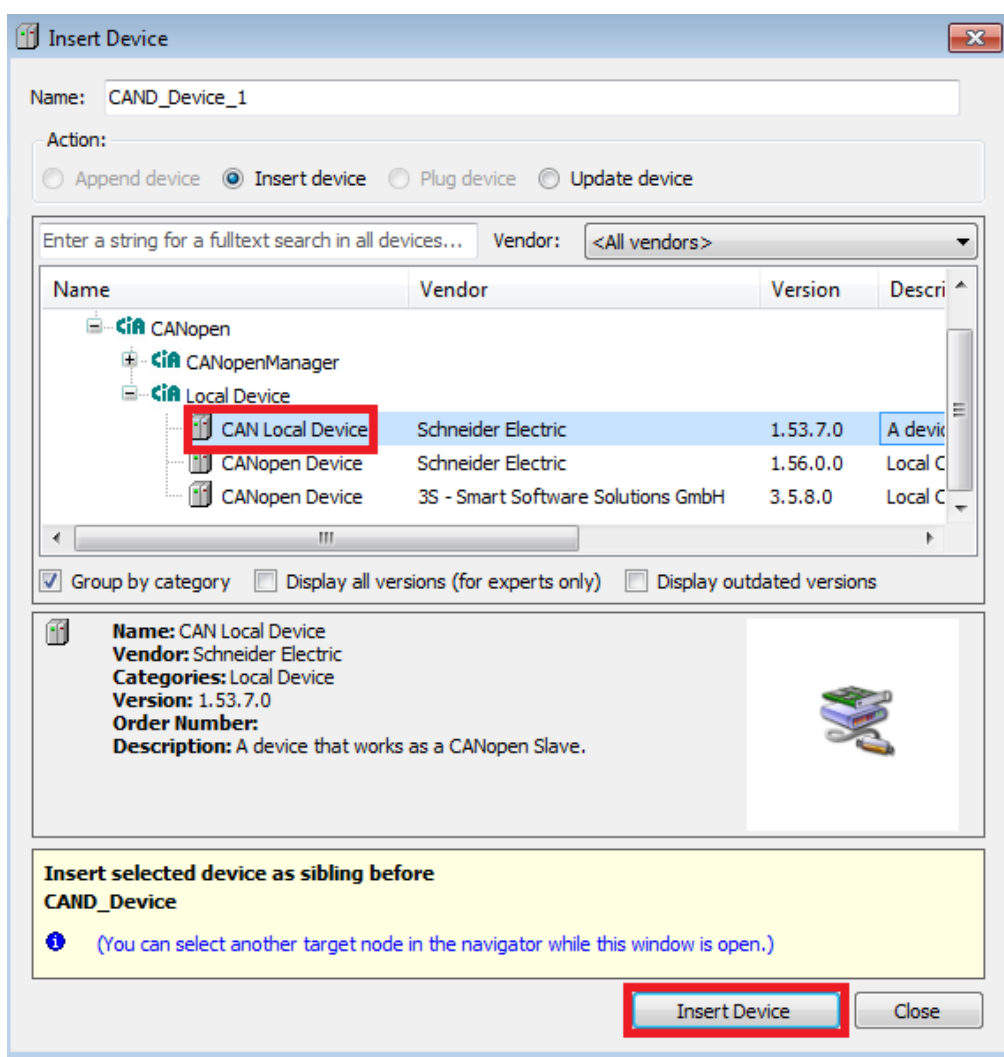
W kolejnym kroku przypiszemy urządzenie, które umożliwi wykorzystanie sterownika PacDrive jako CANOpen Slave. Możemy to zrobić nie zamykając otwartego już okna *Add Device*, poprzez wybieranie ze struktury programu nowododany CAN_Layer. Okno *Add Device* zostanie automatycznie zaktualizowane i umożliwi nam dodanie składników podwładnych wybranemu urządzeniu. Jeżeli okno *Add Device* zostało zamknięte wystarczy kliknąć na moduł CAN_Layer prawym przyciskiem myszy i wybrać z rozwiniętej listy *Add Device*.

Istnieją dwie możliwości konfiguracji protokołu CANopen: dla urządzeń pracujących w oparciu o normę CiA (CAN in Automation) oraz dla innych urządzeń. Innymi słowy. Jeżeli znamy numer profilu urządzenia, który określa interfejs urządzenia logicznego, możemy skorzystać z urządzenia CANopen Device. Gdy takowego numeru nie znamy dodajemy do struktury urządzenie CAN Local Device – to urządzenie będzie konfigurowane w dalszej części instrukcji.

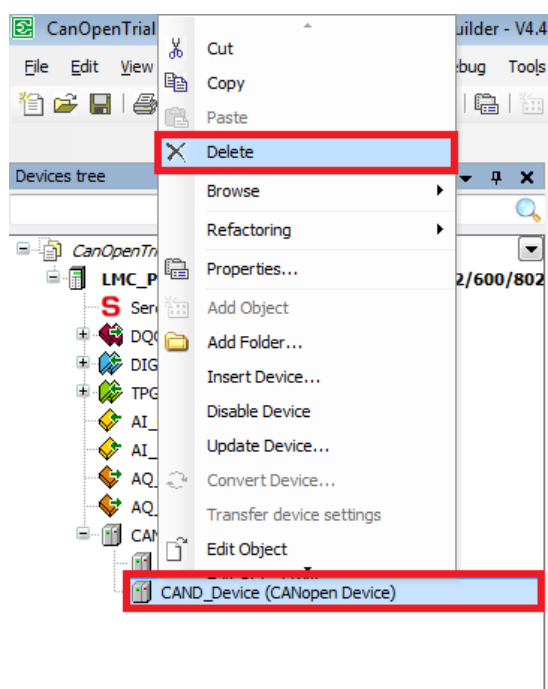
Z nowej listy wybieramy po kolei: *Fieldbusses*>*CANopen*>*Local Device*>*CANopen Device*. Potwierdzamy przyciskiem *Add Device*.



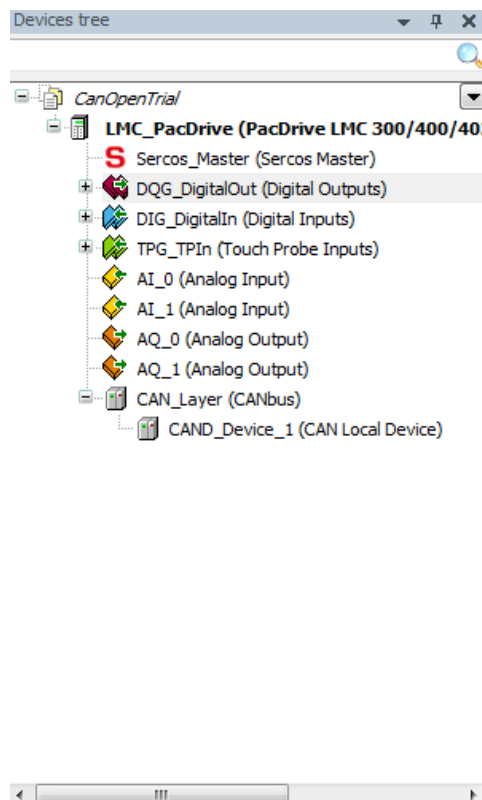
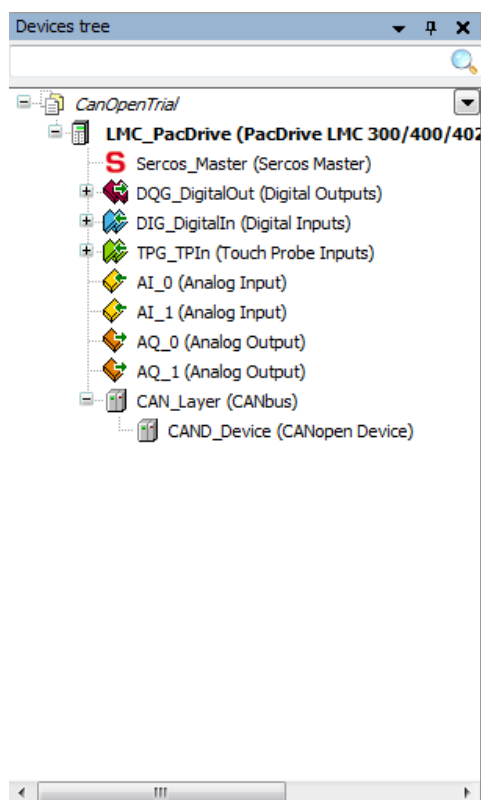
Chcąc dodać urządzenie niekorzystające z dyrektyw CiA, nie zamykając otwartego już okna *Add Device*, w oknie *Device Tree* klikamy dwukrotnie na nowododany *CANopen Device*, a następnie z uaktualnionej listy wybieramy urządzenie *CAN Local Device*, po czym zatwierdzamy wybór przyciskiem *Insert Device*.



Ponieważ moduł *CANopen Device* był nam potrzebny tylko do dodania urządzenia typu Slave, możemy usunąć go ze struktury klikając na niego prawym przyciskiem myszy i wybierając z listy *Delete*.



Gotowa struktura programu w zależności od potrzeb powinna wyglądać jak poniżej:

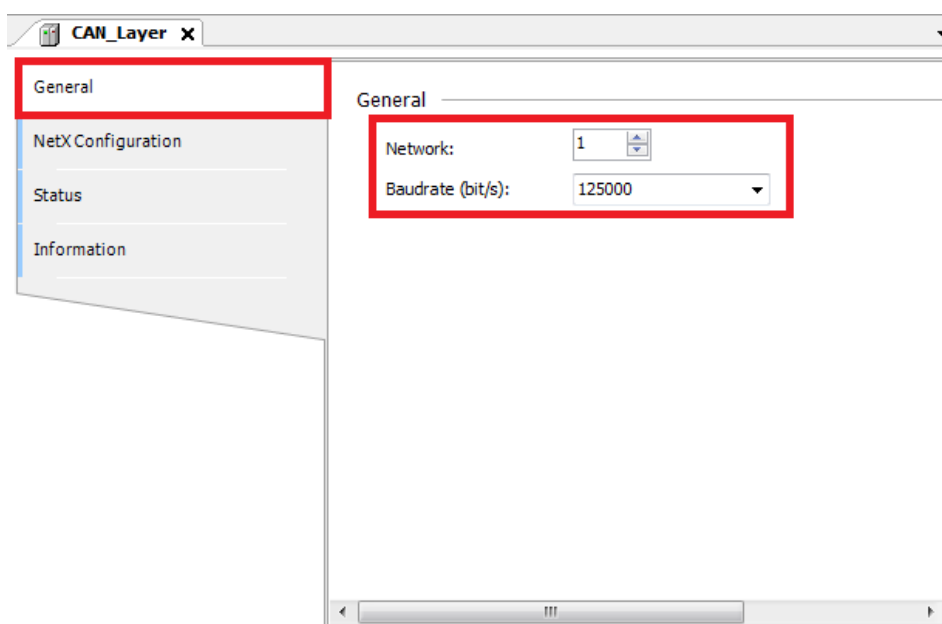


KONFIGUROWANIE MODUŁÓW I URZĄDZEŃ

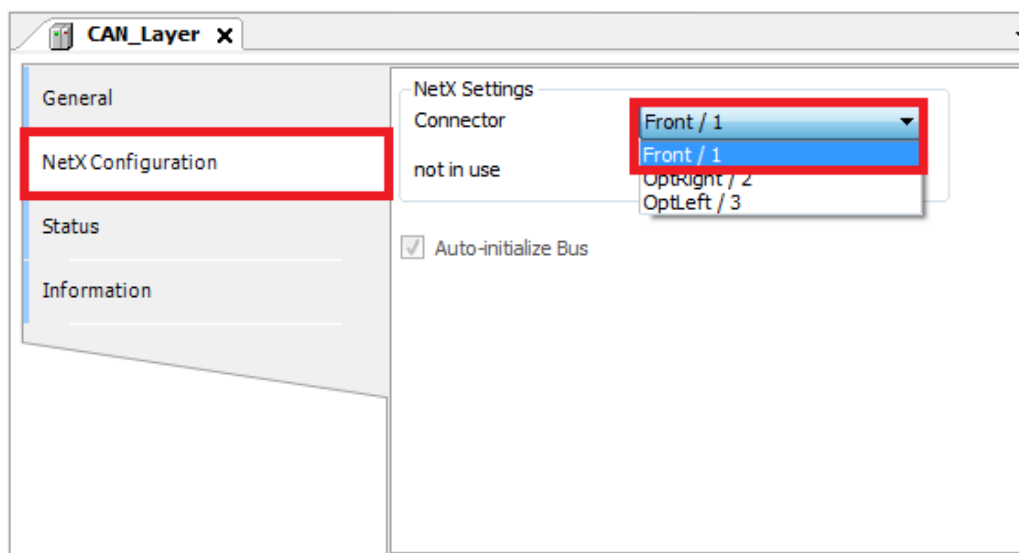
Aby połączenie działało poprawnie należy prawidłowo skonfigurować dodane urządzenie. Aby otworzyć zakładkę z właściwościami obiektu *CAN_Layer* należy dwukrotnie kliknąć na niego w oknie *Devices*, lub po kliknięciu prawym przyciskiem myszy wybrać *Edit Object*.



W grupie *General* zakładki *CAN_Layer* ustawiamy odpowiedni numer urządzenia w sieci CAN, do której podłączony będzie nasz interfejs. W tym przypadku parametr *Network* został ustawiony na 1. Ustawiamy także odpowiedni *Baudrate*, który w tym przykładzie wynosi 125000 bit/s.

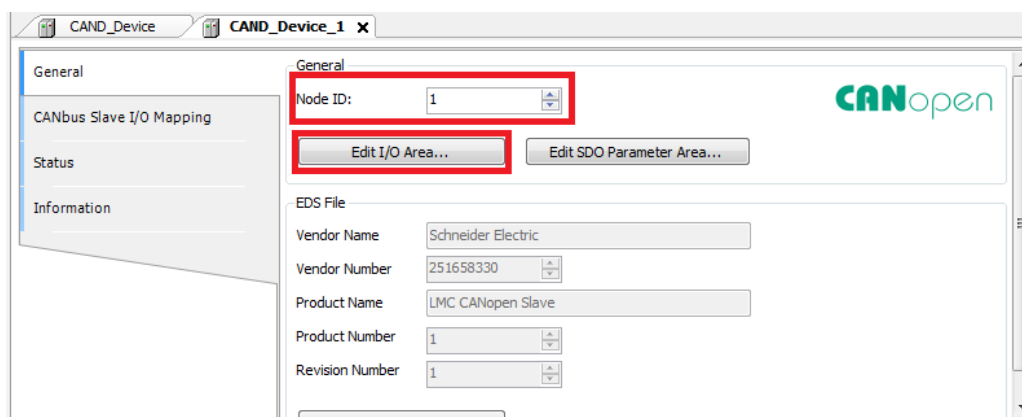


Dalej, w zakładce *NetX Configuration* w polu *Connector* wybieramy złącze, do którego podłączona jest sieć CANopen. W tym przykładzie jest to złącze frontowe.

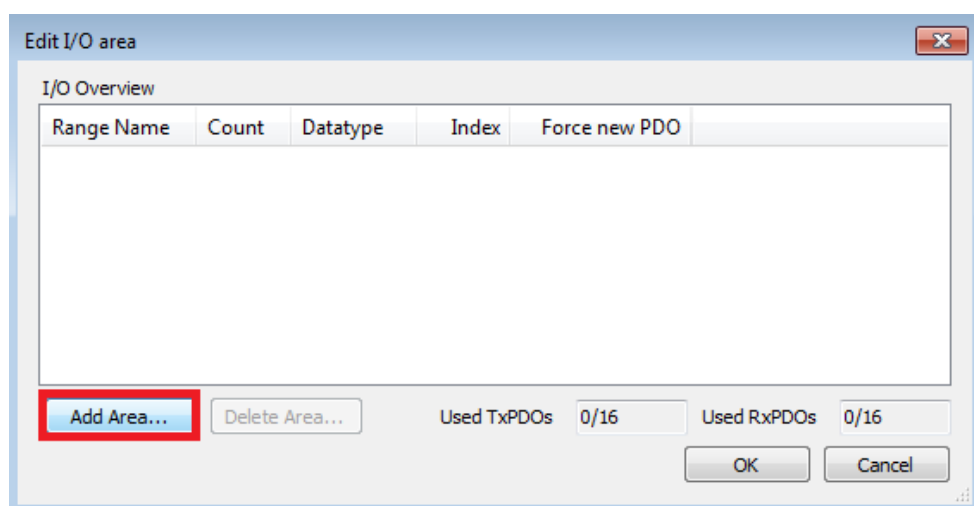


Przechodzimy teraz do konfiguracji modułu o nazwie *CAND_Device_1*. W zakładce *General* ustawiamy *Node ID* na 1. To po wartości parametru *Node ID* CANopen Master będzie identyfikował urządzenie typu Slave. Jeżeli dodaliśmy do struktury urządzenie typu *CANopen Device*, dostępna będzie również druga opcja *Device Profile*, w której należy wybrać odpowiedni profil urządzenia.

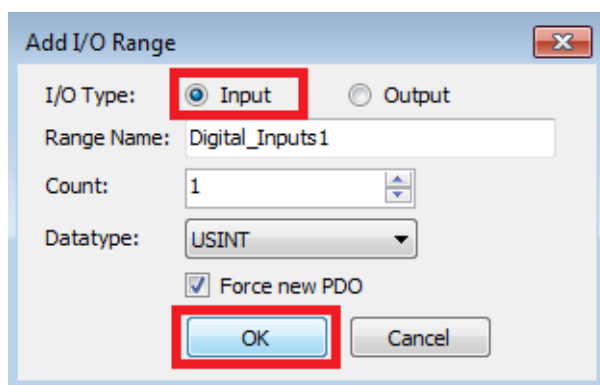
Aby umożliwić wymianę danych między urządzeniami Slave-Master, należy skonfigurować również obszary wejść i wyjść. Robimy to wybierając z zakładki *General* opcję *Edit I/O Area...*



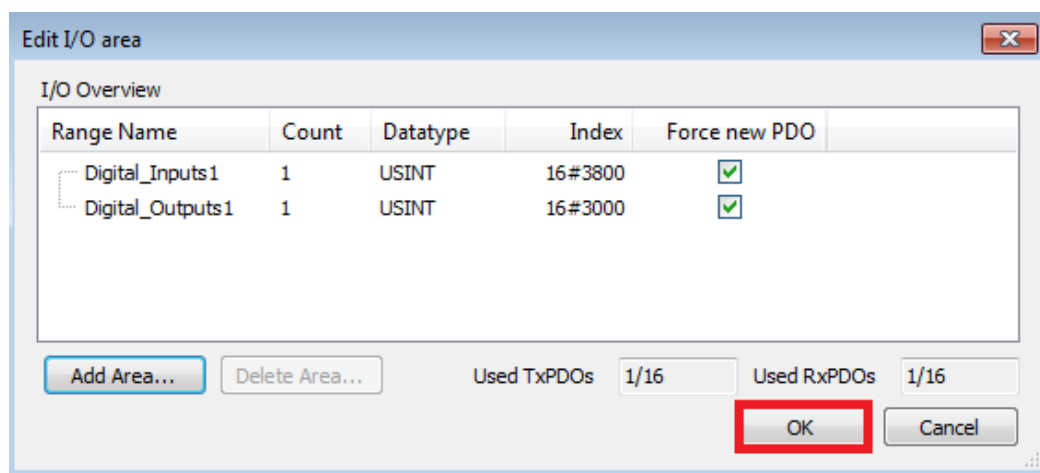
W nowym oknie *Edit I/O Area* wyświetlana jest lista obecnie dodanych do programu zmiennych. Aby dodać nowe zmienne, klikamy przycisk *Add Area...*



Wyświetli się nowe okno, w którym możemy wybrać jej typ, nadać nazwę, ilość, typ danych, a także czy przy dodaniu chcemy wymusić powstanie nowego PDO (ang. *Process Data Object*). Akceptujemy ustawienia domyślne przyciskiem OK, po czym ponawiamy operację zmieniając *I/O Type* na *Output*



Gotową, poniższą listę zmiennych zatwierdzamy przyciskiem OK.



Ostatnim elementem, który należy skonfigurować w programie SoMachine Motion Logic Builder jest sposób odświeżania wartości z modułów podległych pod CANopen Slave. Parametr ten zmieniamy oddzielnie w każdym module lub nadrzędnie dla wszystkich modułów i zmiennych wykorzystywanych w projekcie w konfiguracji sterownika LMC.

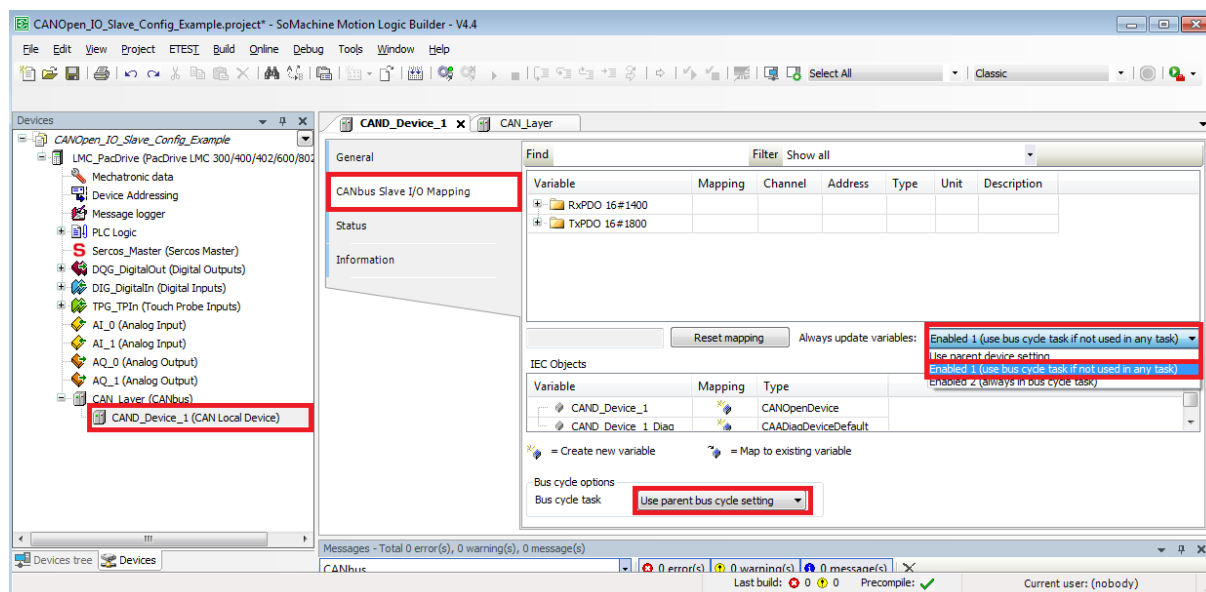
Aby zmienić sposób odświeżania oddzielnie dla każdego modułu klikamy na niego w drzewku programu i otwieramy jego okno parametrów. Wybieramy zakładkę *CANbus Slave I/O Mapping* i zmieniamy parametr *Always update variables*. Parametr ten może przyjmować jedną z trzech wartości:

- *Use parent device setting* – aktualizuje zmienne zgodnie z ustawieniami urządzenia nadrzędnego, czyli w tym przypadku modułu CANbus.
- *Enabled 1 (use bus cycle task if not used in any task)* - jeżeli zmienne nie są używane w żadnym zadaniu to są aktualizowane zgodnie z tzw. *bus cycle task*. Zadanie *bus cycle task* konfiguruje się w ustawieniach sterownika LMC.
- *Enabled 2 (always in bus cycle task)* - zmienne są aktualizowane w każdym cyklu zadania *bus cycle task*, niezależnie od tego, czy są używane lub mapowane.

Na potrzeby tej instrukcji ustawmy parametr *Always update variables* w każdym module wymiany danych na *Enabled 1 (use bus cycle task if not used in any task)*. Pozwoli to na odczyt wartości zmiennych bez używania ich w programie.

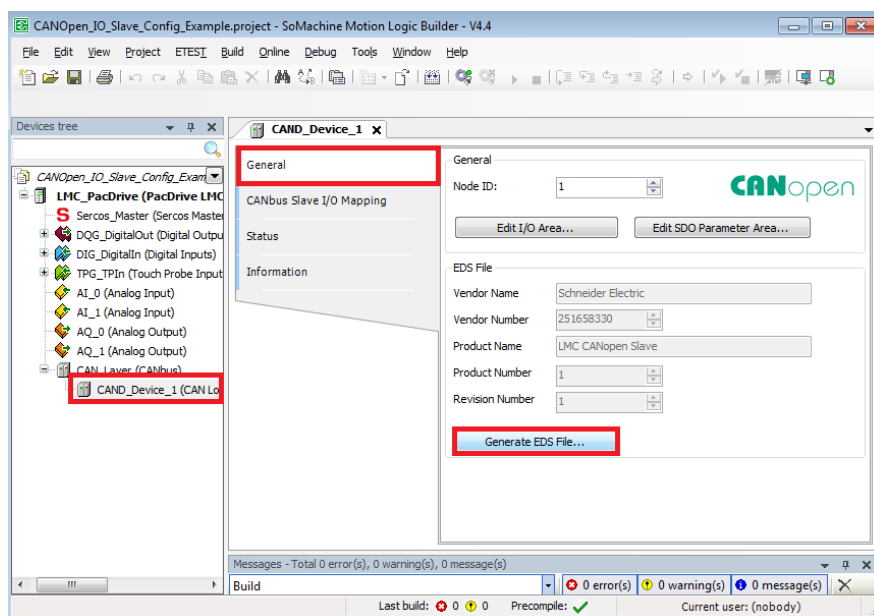
Zmienne będą odświeżane zgodnie z okresem wywołania *bus cycle task*, które możemy ustawić w parametrach kontrolera LMC, w zakładce *PLC settings*. W tym miejscu będą wyświetlane wszystkie zadania procesora, które są utworzone w programie. Wybierając opcję *<unspecified>* wybierane będzie najwolniejsze zadanie cykliczne. Dla potrzeb tej instrukcji nie musimy zmieniać tego parametru.

W zakładce *PLC settings* możemy także ustawić nadrzędną wartość dla parametru *Always update variables*. Wartość ta dotyczy wszystkich zmiennych oraz wejść i wyjść zdefiniowanych w drzewku programu. Dla potrzeb tej instrukcji nie musimy zmieniać tego parametru.

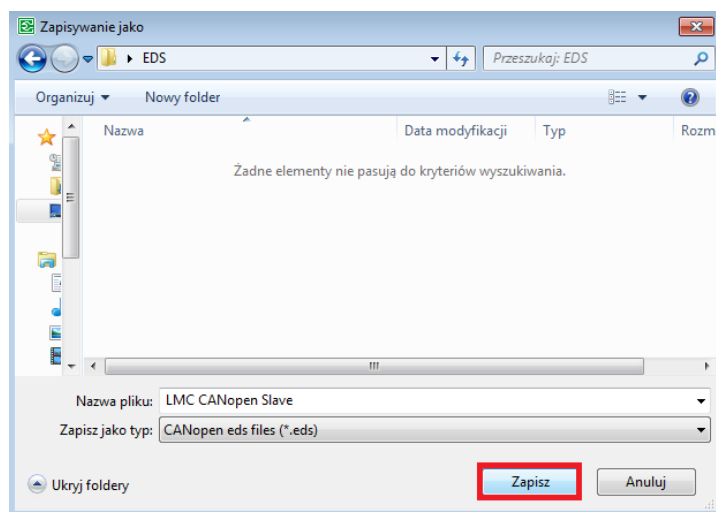


ZAPISYWANIE KONFIGURACJI DO PLIKU EDS

Ważnym krokiem jest wyeksportowanie pliku EDS, którym będziemy się posługiwać przy konfigurowaniu urządzenia typu Master. W tym celu przechodzimy do ustawień modułu CAND_Device_1 i w zakładce *General* wybieramy opcję *Generate EDS File...*



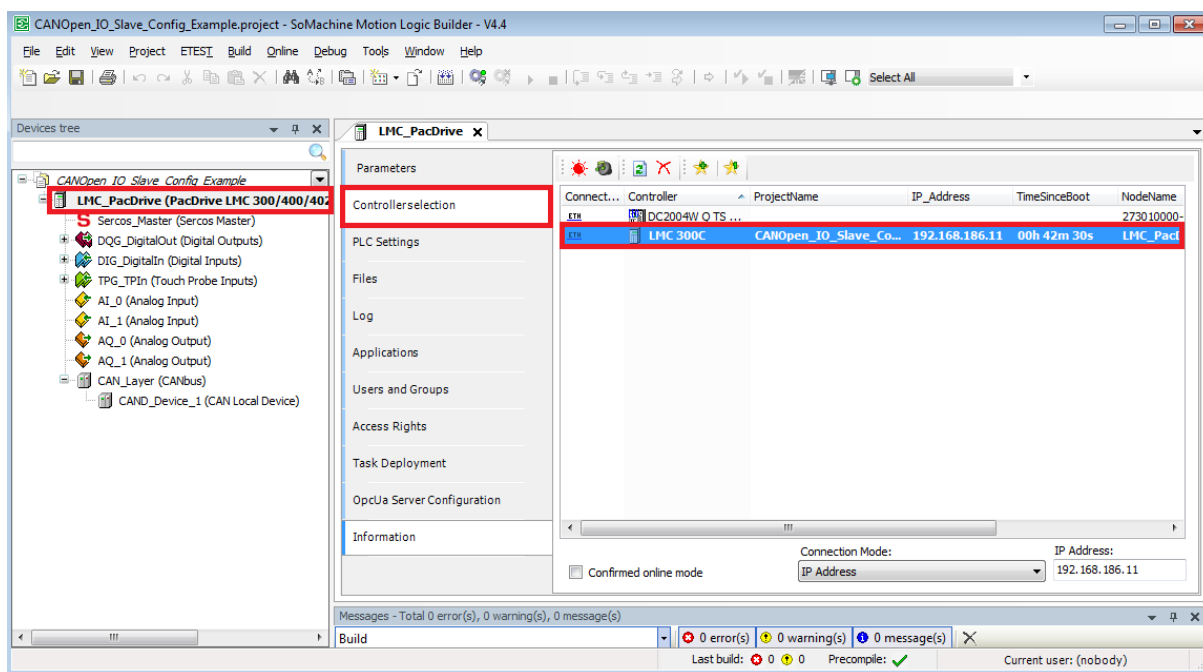
Pojawi się nowe okno, w którym wybieramy lokalizację dla naszego pliku o rozszerzeniu .eds a także możemy nadać mu jego nazwę. Wybór zatwierdzamy przyciskiem *Zapisz*.



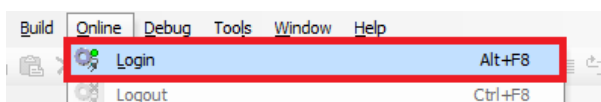
Po zakończeniu konfiguracji można wgrać program na sterownik PacDrive 3 LMC w celu późniejszych testów działania komunikacji CANopen.

WGRYWANIE PROGRAMU DO KONTROLERA

W pierwszej kolejności musimy określić sterownik, do którego chcemy wgrać program - otwieramy ustawienia urządzenia LMC, przechodzimy do zakładki *Controller selection* i klikamy dwukrotnie na wybrany sterownik, który powinien zostać pogrubiony.




Aby wgrać projekt na kontroler i połączyć się z nim, z paska programu wybieramy opcję Online/Login lub klikamy na ikonę widoczną na pasku szybkiego uruchamiania.

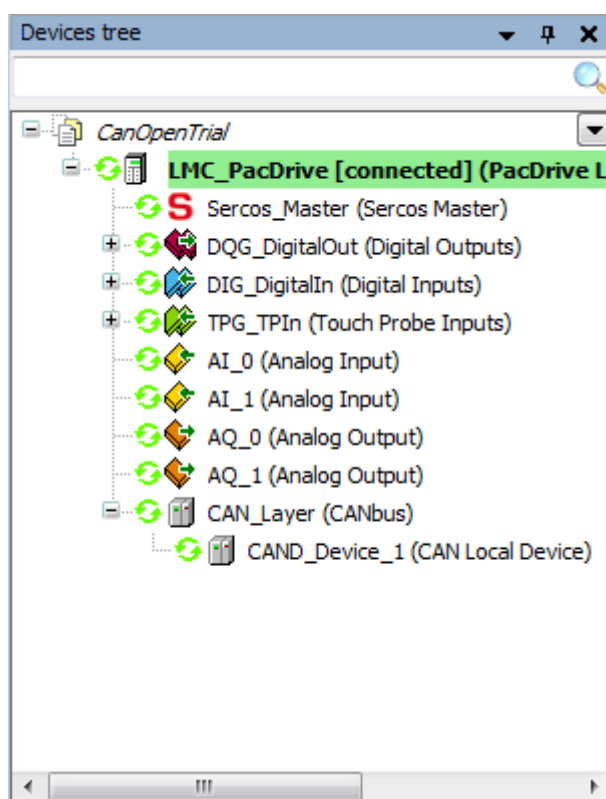




Po zatwierdzeniu komunikatów następuje wgranie programu sterującego do kontrolera. Po wgraniu programu może być wymagany restart sterownika (*cold reset*), aby wprowadzone zostały wszystkie modyfikacje.

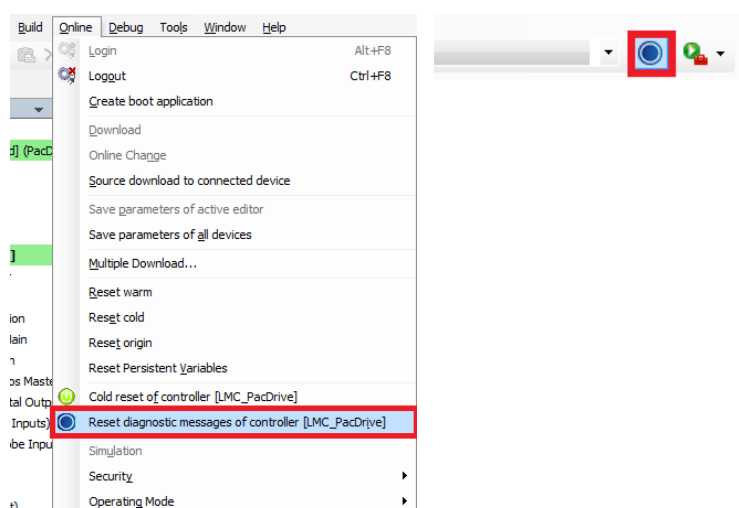
Po zakończonej procedurze restartu należy uruchomić program poprzez przycisk Start znajdujący się na pasku szybkiego uruchamiania, lub poprzez skrót klawiszowy F5. Jeżeli urządzenie CANopen Slave jest już prawidłowo skonfigurowane to komunikacja powinna zostać ustanowiona.

Uwaga! Aby komunikacja działała poprawnie, moduł pracujący jako CANopen Master musi być uruchomiony i połączony ze sterownikiem.

Jeżeli konfiguracja została poprawnie przeprowadzona i połączenie zostało pomyślnie nawiązane, obok struktury urządzeń znajdzie się ikona  świadcząca o braku błędów.



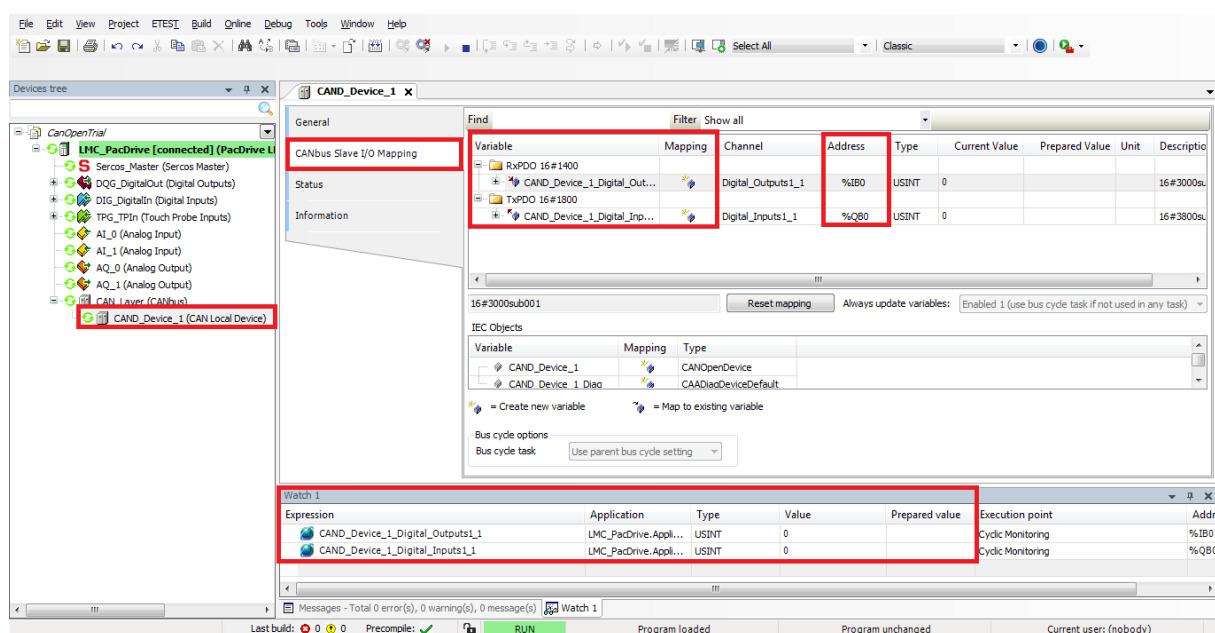
Jeżeli zaś obok któregoś z komponentów pojawi się ikona , to informuje ona o jego błędnej konfiguracji. Należy ponownie sprawdzić ustawienia urządzeń i ustawić poprawnie parametry. Przy ponownej próbie połączenia, gdy w sterowniku znajdują się niezatwierdzone błędy, komunikacja z urządzeniami zewnętrznymi może nie działać poprawnie. Błędy te sygnalizowane są poprzez wykrzyknik  obok kontrolera LMC na drzewku urządzeń. Aby usunąć błędy należy wybrać opcję z paska zadań *Online/Reset diagnostic messages of controller*, lub odpowiednią ikonę z paska szybkiego uruchamiania.



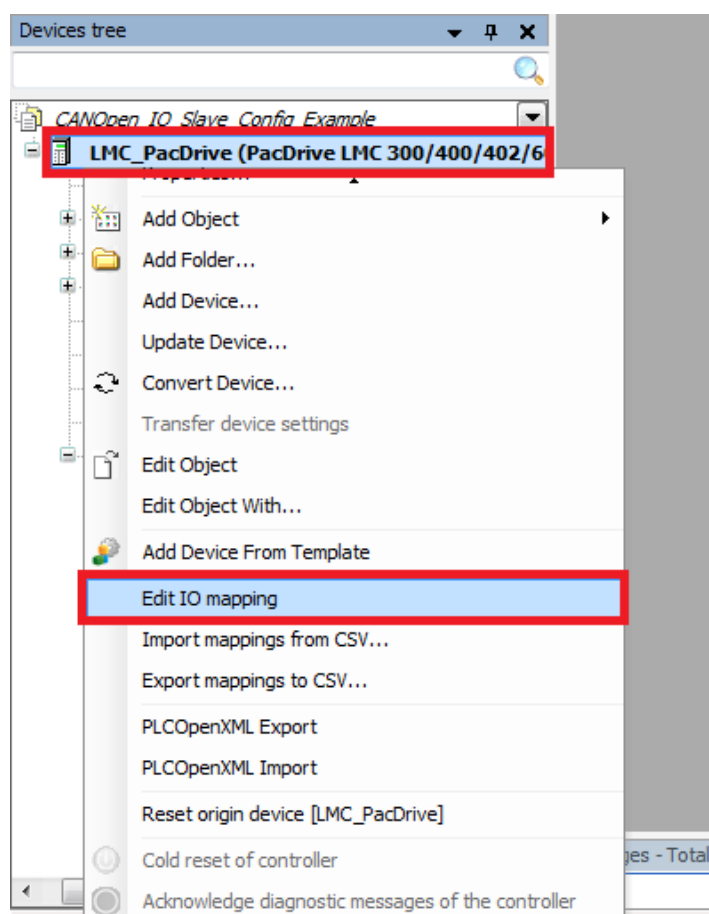
MAPOWANIE IO

Odczytanie wartości zmiennych wejściowych lub sterowanie zmiennymi wyjściowymi może odbywać się na kilka sposobów:

- Ręcznie poprzez wejście w zakładkę *CANbus Slave I/O Mapping* wybranego modułu – odczytujemy wartość w kolumnie *Current value* a wpisujemy nową wartość wprowadzając ją w kolumnie *Prepared value* i następnie wybierając *Debug/Write values* lub naciskając Ctrl+F7 przesyłamy je do kontrolera.
- Wykorzystując adres wymienianych danych – adresy kolejnych zmiennych są podane w zakładce *CANbus Slave I/O Mapping* dla każdego z modułów. W programie sterującym możemy odwoływać się do zmiennych wykorzystując ich adresy. Dodatkowo możemy odwoływać się do zgrupowanych zmiennych – np. zmienne wejściowe pogrupowane są w bajty, co umożliwia odczyt 8 zmiennych binarnych w postaci dziesiętnej.
- Wykorzystując nazwę zmapowanych danych – znacznie wygodniejsze podczas programowania jest odwoływanie się do danych z komunikacji po ich zmapowanych nazwach. Aby wykonać mapowanie konieczne jest rozłączenie się ze sterownikiem i wprowadzenie nazw zmiennych w pola kolumny *Variable*. Zmienne, których nazwy wprowadzamy tworzone są dynamicznie (nie wymagają wcześniejszej deklaracji).



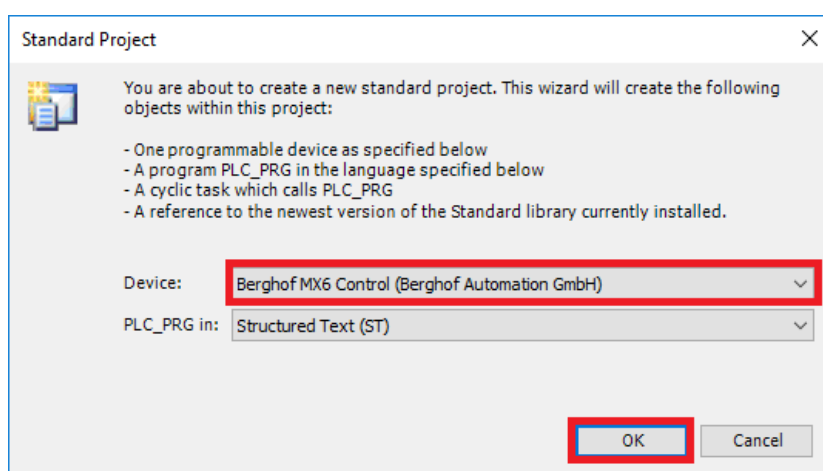
Pełną listę mapowań wejść/wyjść można otworzyć klikając prawym przyciskiem myszy na kontroler LMC i wybierając Edit IO Mapping.



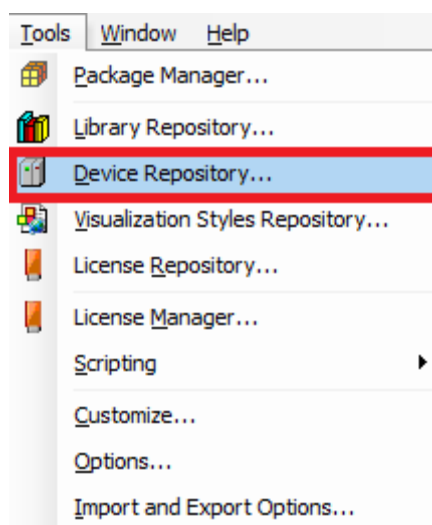
Konfiguracja sterownika Astraada One jako CANopen Master w środowisku CODESYS

BUDOWANIE STRUKTURY PROGRAMU

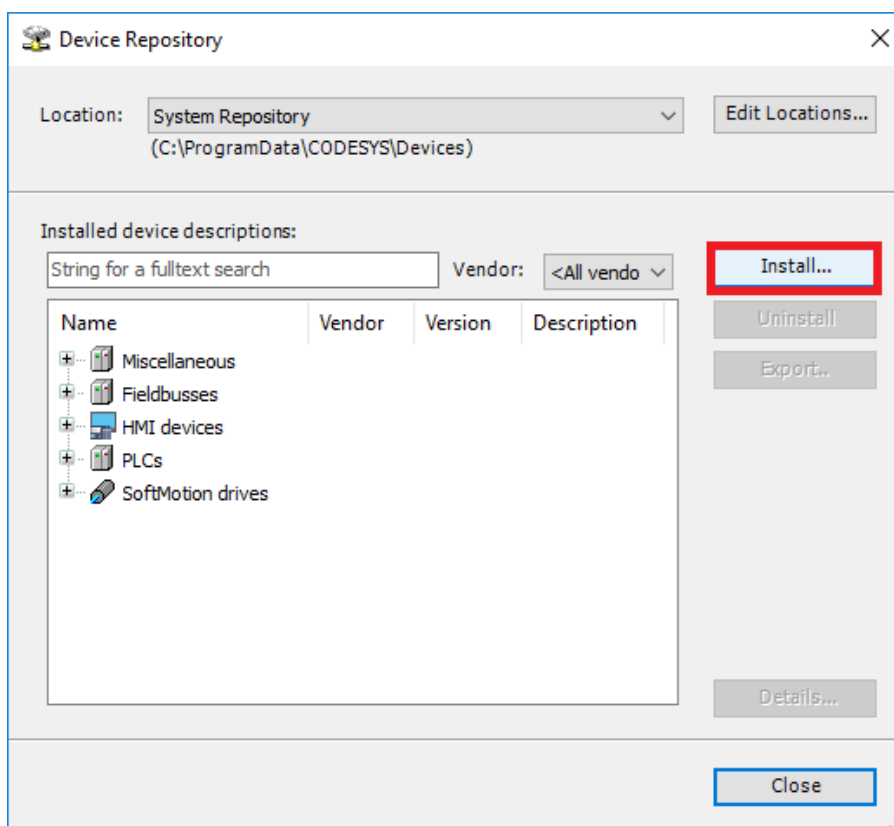
Po utworzeniu nowego projektu typu *Standard* w programie CODESYS w oknie, które się pojawi wybieramy urządzenie Berghof MX6 Control (Berghof Automation GmbH) oraz odpowiedni dla nas język programowania, w tym przypadku ST.



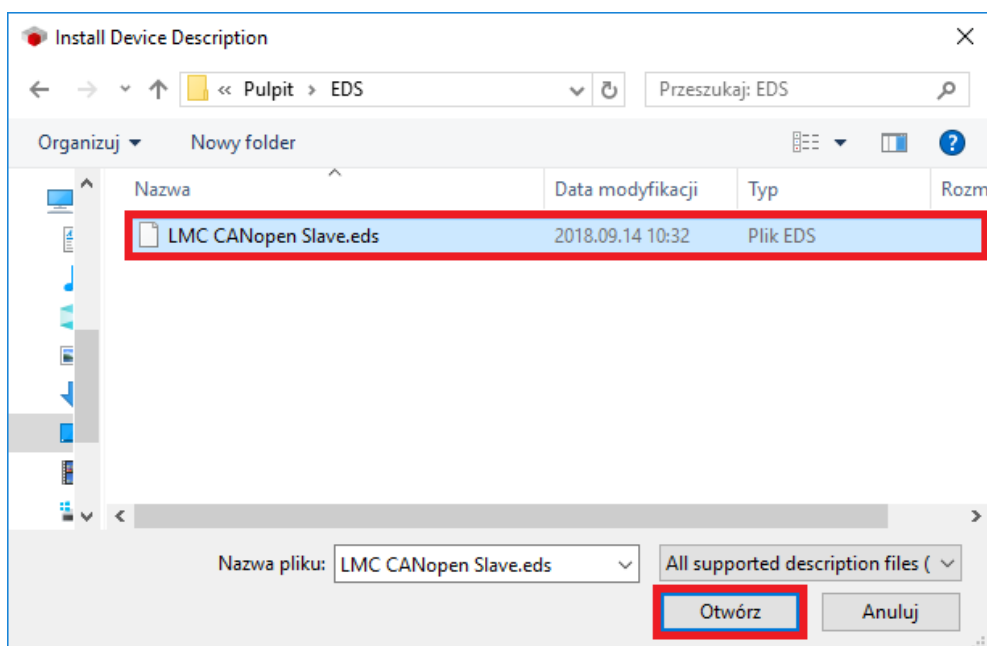
Zostanie zbudowana podstawowa struktura programu obsługująca urządzenie Astraada One. Aby skonfigurować ją jako CANopen Master, do biblioteki urządzeń dodamy sterownik LMC PacDrive. W tym celu, z paska narzędzi wybieramy zakładkę *Tools>Device Repository*.



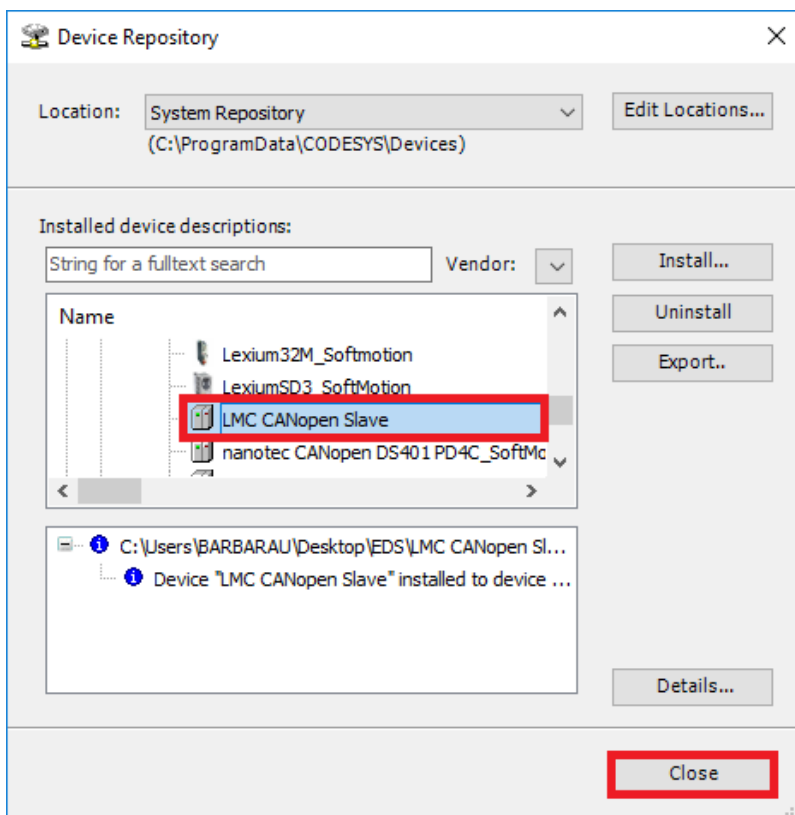
Pojawi się okno z listą wszystkich urządzeń zainstalowanych do tej pory. Aby dodać nowe, klikamy przycisk *Install...*



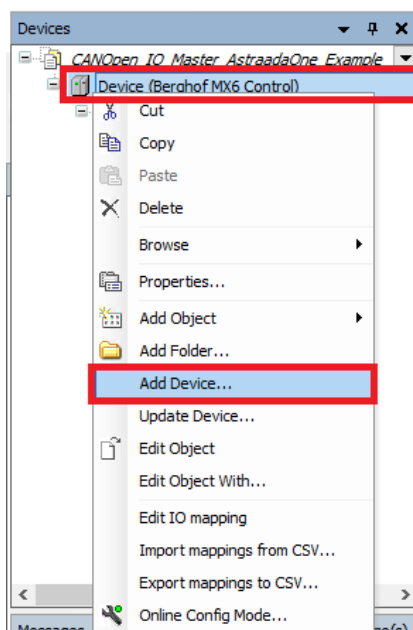
W nowym oknie wybieramy ścieżkę dostępu i plik o rozszerzeniu .eds wyeksportowany podczas konfiguracji sterownika PacDrive LMC jako CANopen Slave, co zostało opisane w poprzednim rozdziale informatora. Zatwierdzamy przyciskiem *Otwórz*.



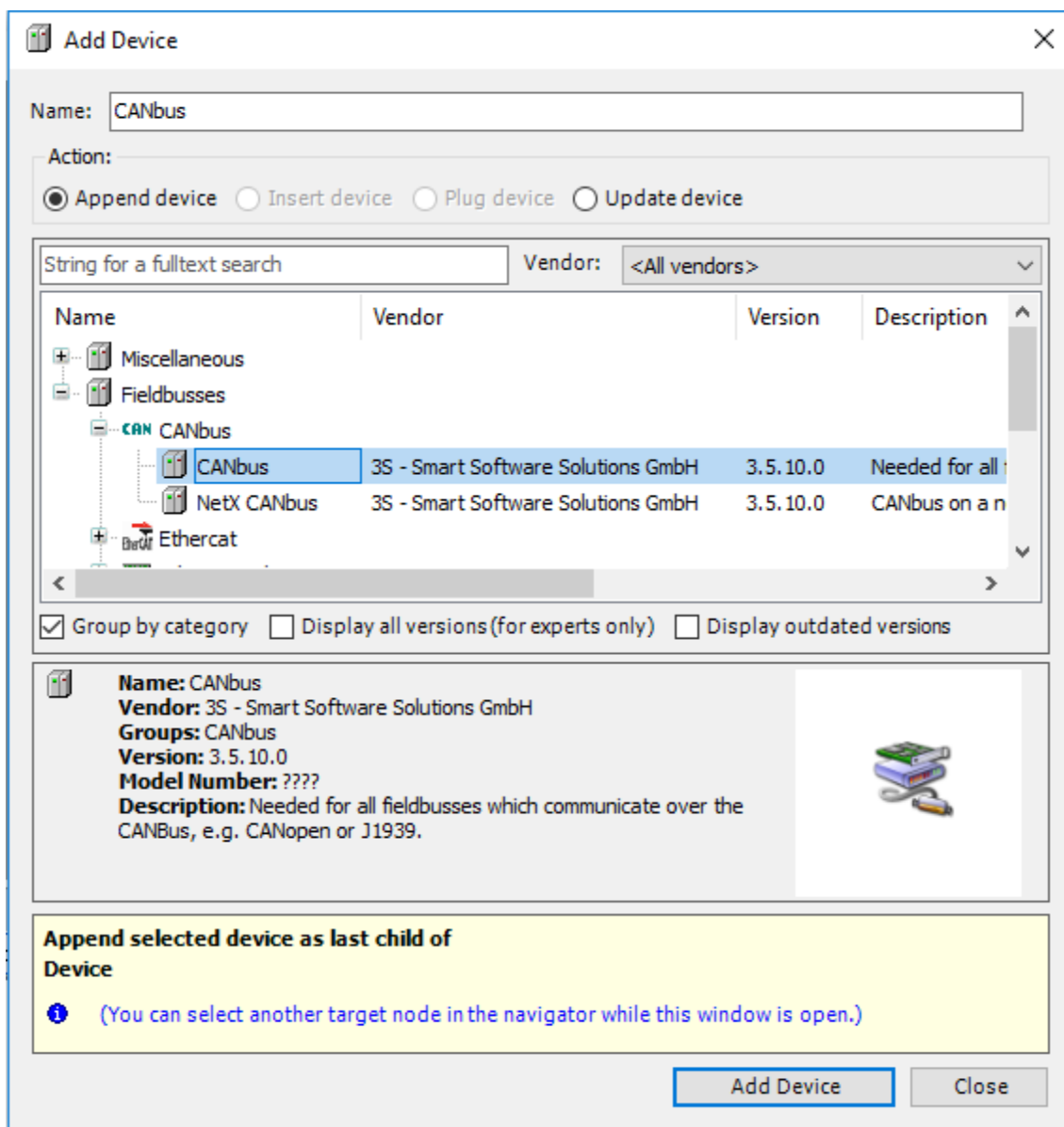
Lista urządzeń zostanie zaktualizowana i będziemy w stanie znaleźć na niej urządzenie LMC. Okno zamykamy przyciskiem *Close*.



W następnym kroku należy dodać moduł obsługujący komunikację po protokole CANopen. W tym celu, na drzewku urządzeń, klikamy prawym przyciskiem myszy na urządzenie *Device (Berghof MX6 Control)* i z rozwiniętych opcji wybieramy *Add device...*

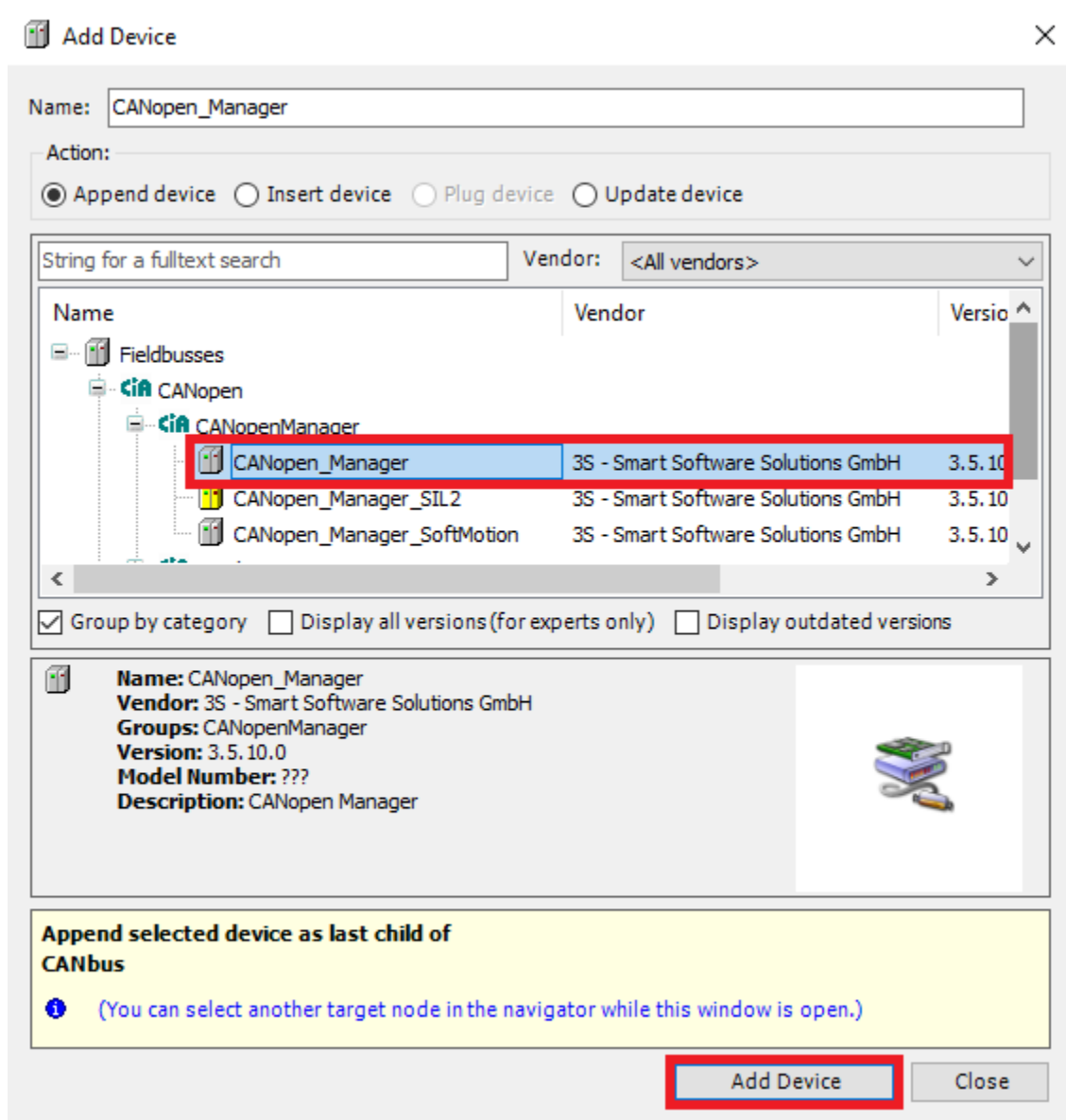


W nowym oknie rozwijamy po kolei *Fieldbuses*>*CANbus* i wybieramy *CANbus*, zatwierdzając urządzenie przyciskiem *Add Device*.

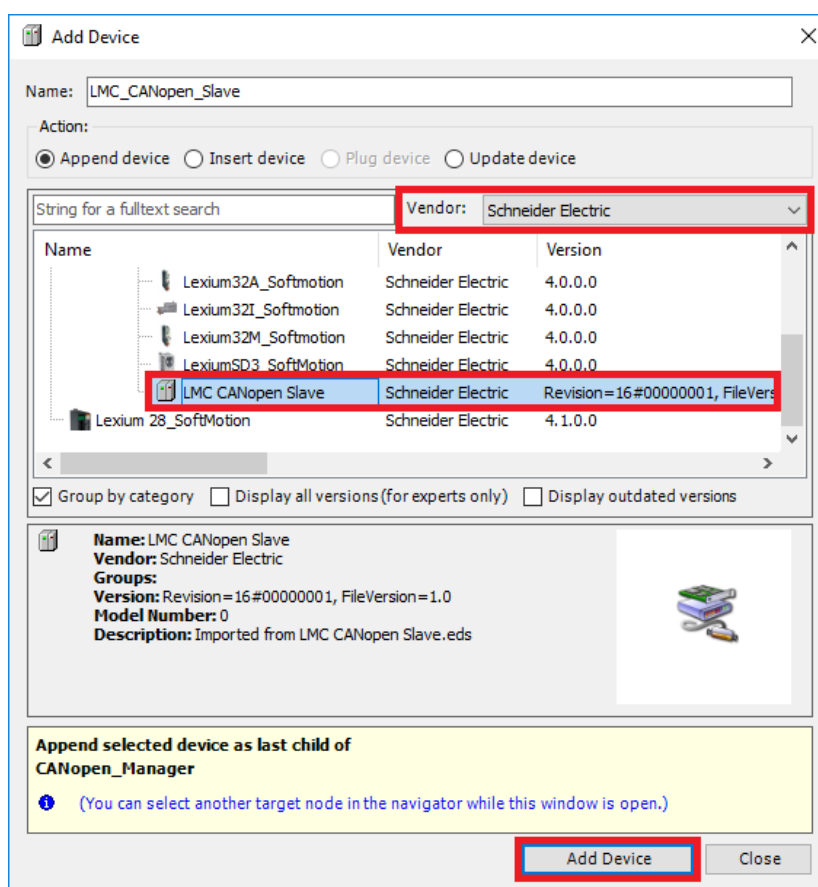


Dalej przypiszemy urządzenie, które umożliwi wykorzystanie sterownika Astraada One jako CANopen Master. Możemy to zrobić nie zamykając otwartego już okna *Add Device*, poprzez wybranie z drzewka programu *Device tree* nowododanego obiektu typu CANbus. Okno *Add Device* zostanie automatycznie zaktualizowane i umożliwi nam dodanie składników podrzędnych do zaznaczonego typu urządzenia. Jeżeli okno *Add Device* zostało zamknięte wystarczy kliknąć na moduł CANbus prawym przyciskiem myszy i wybrać z rozwiniętej listy *Add Device*.

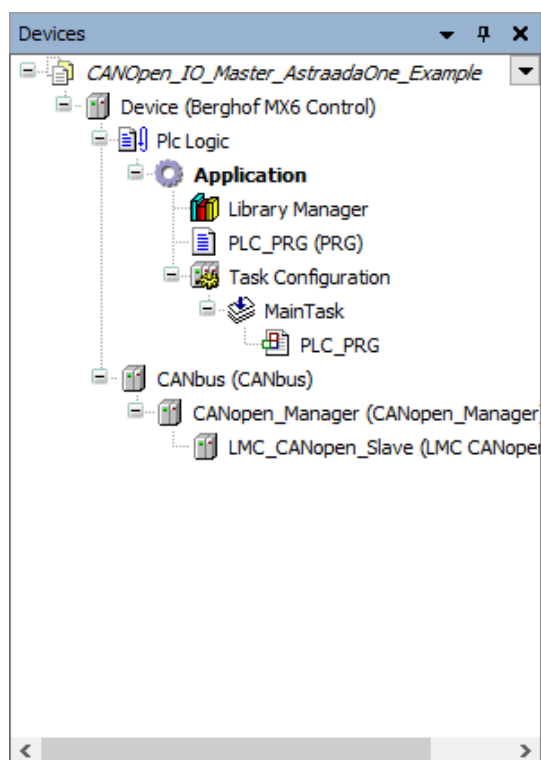
Z nowej listy urządzeń wybieramy *Fieldbusses>CANopen>CANopenManager>CANopen Manager*.



Odświeżamy listę wybierając z drzewka urządzeń (*Device tree*) nowododany komponent, a następnie przechodzi do: *Fieldbusses>CANopen>RemoteDevice*. Aby ułatwić wyszukanie odpowiedniego modułu typu Slave, w polu *Vendor*: możemy zaznaczyć producenta, którego produkty chcemy wyświetlać. Lista zostanie ograniczona i wybieramy z niej moduł LMC CANopen Slave. Zatwierdzamy przyciskiem *Add Device*.

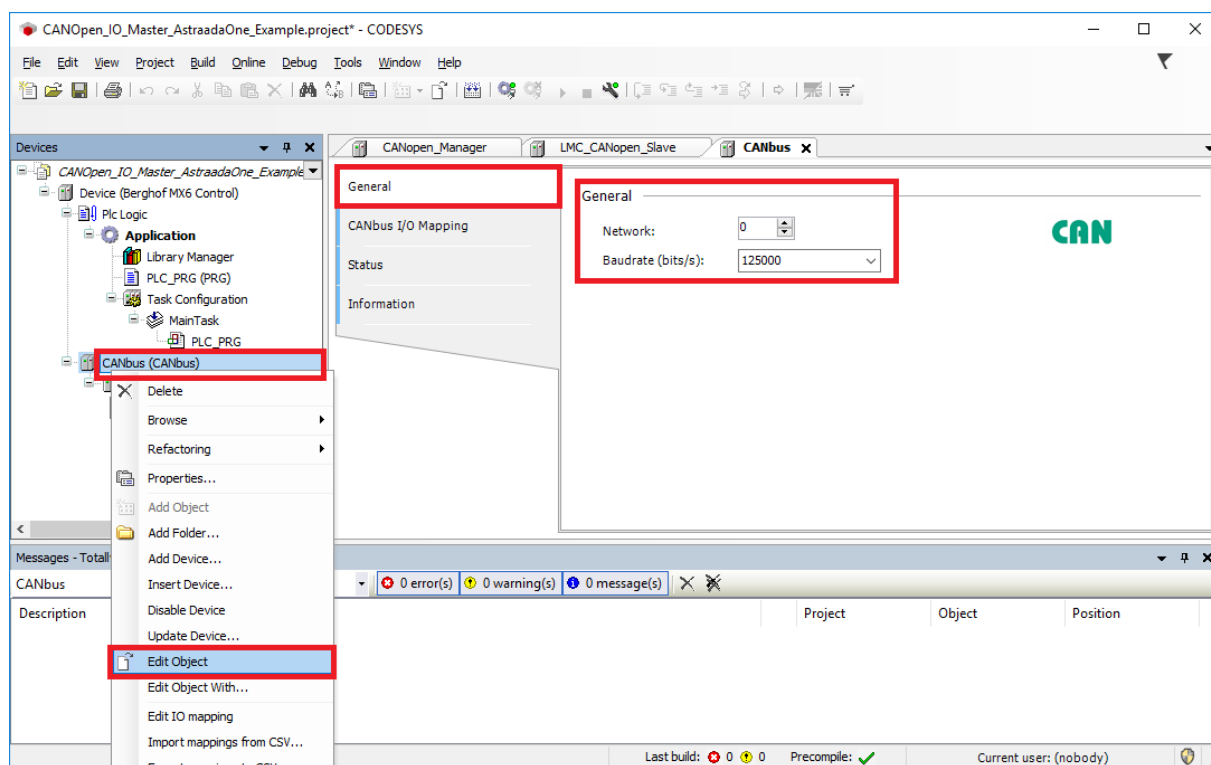


Gotowa struktura programu wygląda następująco:

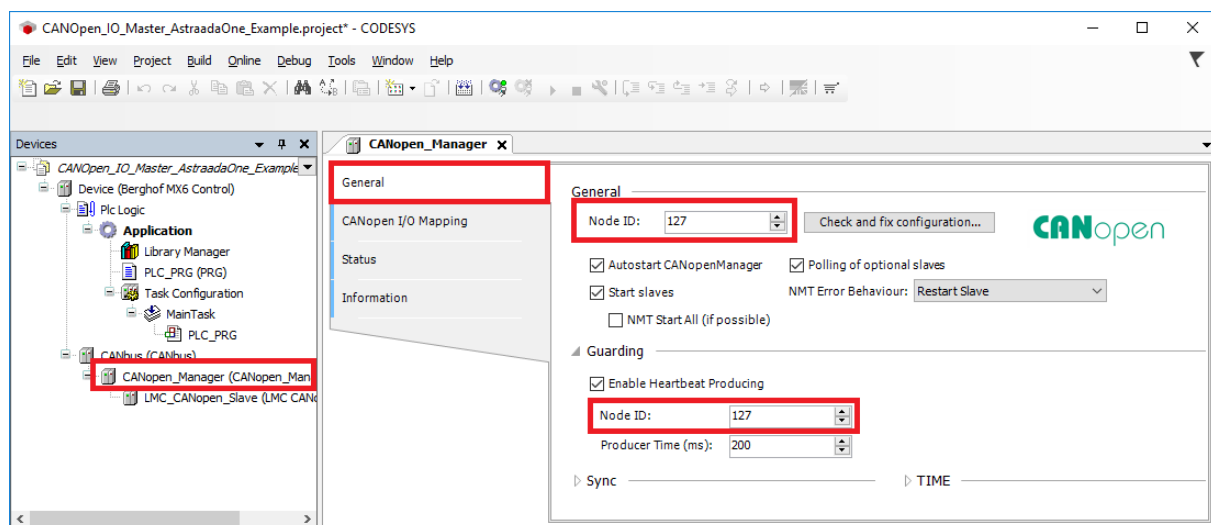


KONFIGUROWANIE MODUŁÓW I URZĄDZEŃ

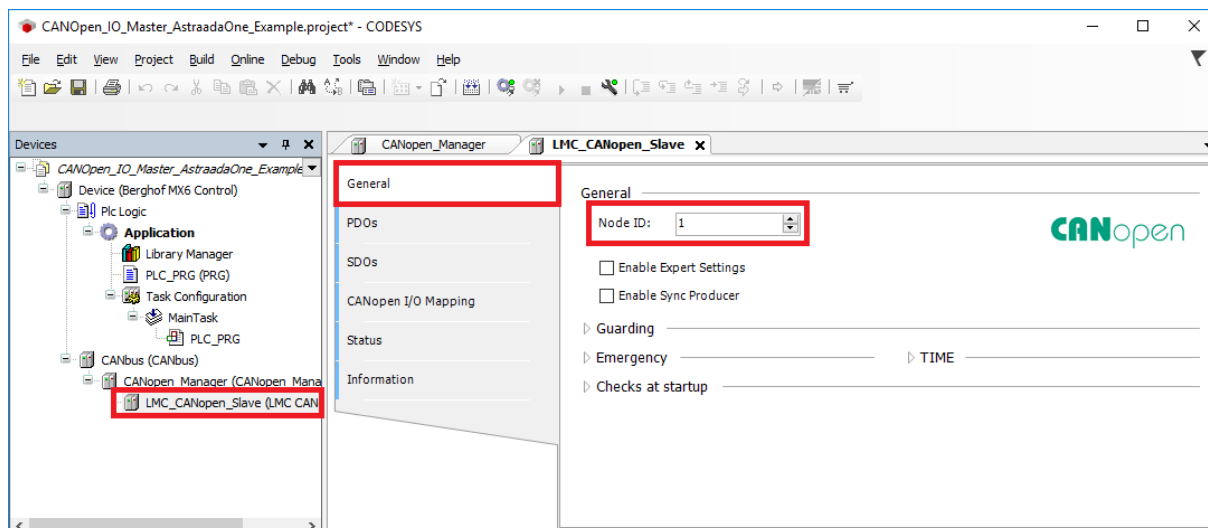
Rozpoczynamy konfigurację od modułu CANbus. Do jego ustawień wchodzimy klikając dwukrotnie na jego nazwę w drzewku urządzeń, lub klikając na niego prawym przyciskiem myszy i z rozwiniętej listy wybierając *Edit Object*. W grupie *General* ustawiamy *Baudrate* oraz *Network*. W tej instrukcji *Baudrate* wynosi 125000 bit/s, a numer *Network* to 0. Pamiętajmy, że jest to numer indywidualny dla każdego urządzenia i przy bardziej rozbudowanych programach nie może się on powtarzać.



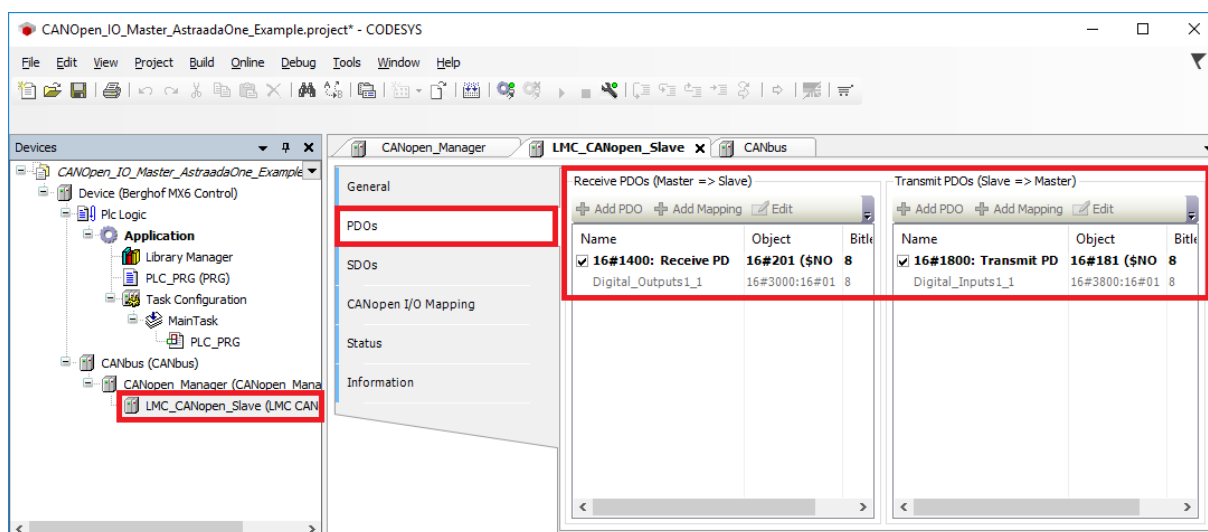
Następnie przechodzimy do ustawień modułu CANopen Manager, gdzie w zakładce *General* możemy skonfigurować obsługę protokołu komunikacji. Dla tej instrukcji nie będziemy w tym miejscu niczego zmieniać, jedynie należy upewnić się że *Node ID* w polu *General* jest równy temu podanemu w polu *Guarding*.



Jako ostatnie konfigurujemy urządzenie LMC_CANopen_Slave, gdzie w grupie *General* zmieniamy *Node ID* na wartość ustawioną w poprzedniej części informatora (w naszym przypadku *Node ID* wynosiło 1).



Przechodząc do zakładki *PDOs* możemy zobaczyć, że wraz ze wgraniem pliku EDS, załadowane zostały także informacje na temat wymiany danych między urządzeniem Slave a Master.

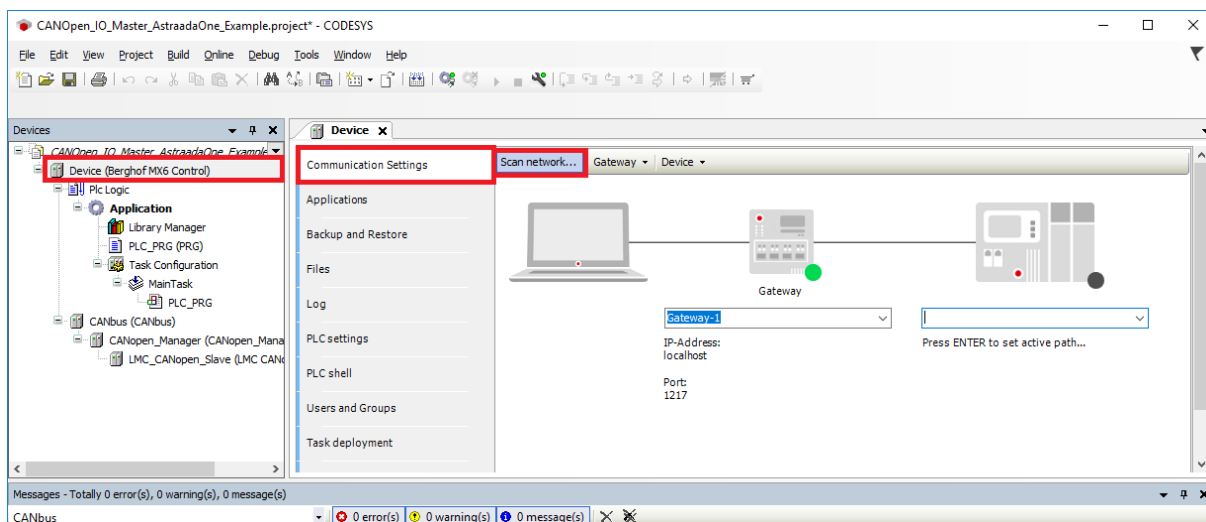


MAPOWANIE IO

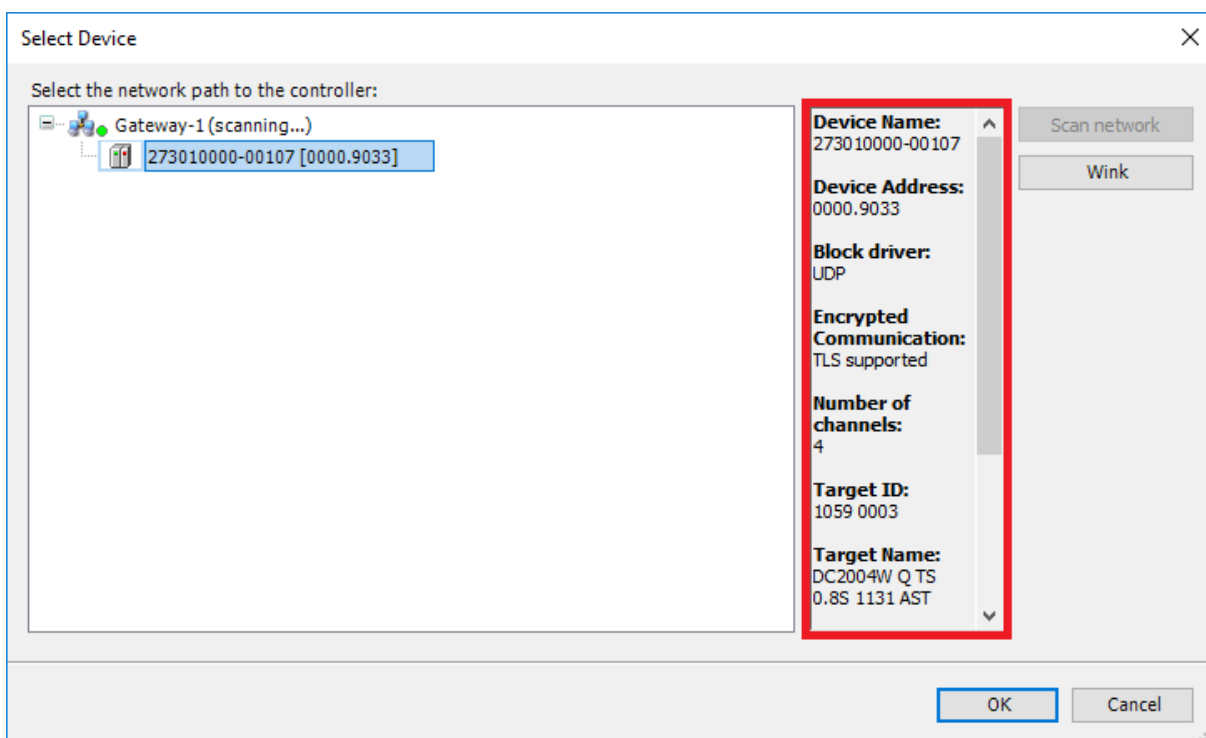
Odczytanie wartości zmiennych wejściowych lub sterowanie zmiennymi wyjściowymi odbywa się dokładnie w ten sam sposób jaki został opisany w poprzedniej części informatora.

WGRYWANIE PROGRAMU DO KONTROLERA

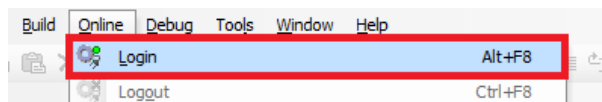
W celu połączenia się ze sterownikiem wybieramy ze struktury programu urządzenie *Device*, a następnie w grupie *Communication Settings* klikamy w przycisk *Scan network*.



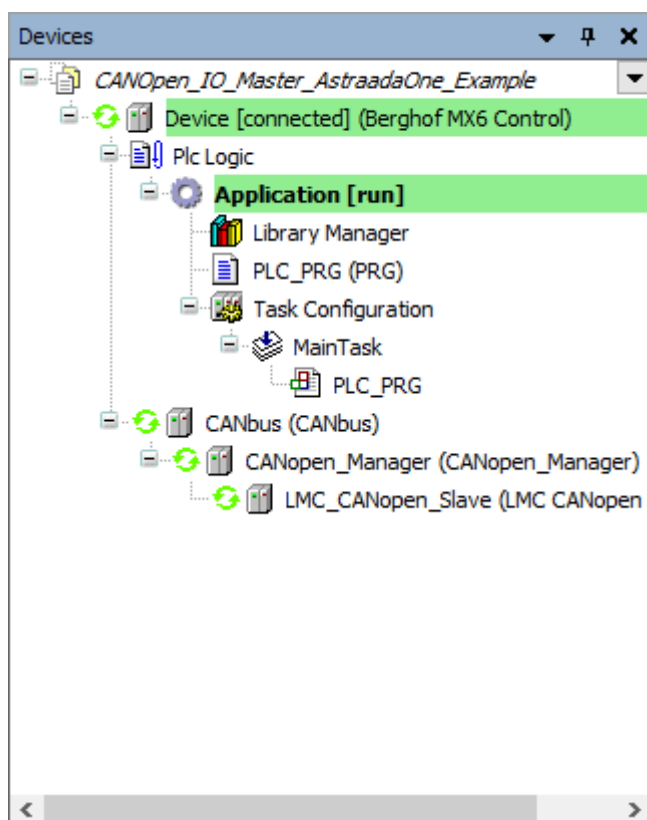
Jeżeli nasz sterownik jest podłączony do sieci, powinien on być widoczny na liście urządzeń, która wyświetli się wraz z nowym oknem *Select Device*. Zaznaczając dowolne urządzenie na liście, w polu obok pojawią się dodatkowe informacje z nim związane, jak np. *Target Name*. W ten sposób możemy upewnić się, że łączymy się z odpowiednim urządzeniem. Wybór potwierdzamy przyciskiem *OK*, lub dwukrotnym kliknięciem w nazwę komponentu.



Aby wgrać projekt na kontroler i połączyć się z nim, z paska programu wybieramy opcję Online/Login lub klikamy na ikonę widoczną na pasku szybkiego uruchamiania.



Jeżeli konfiguracja jest poprawna, struktura programu powinna teraz wyglądać następująco:



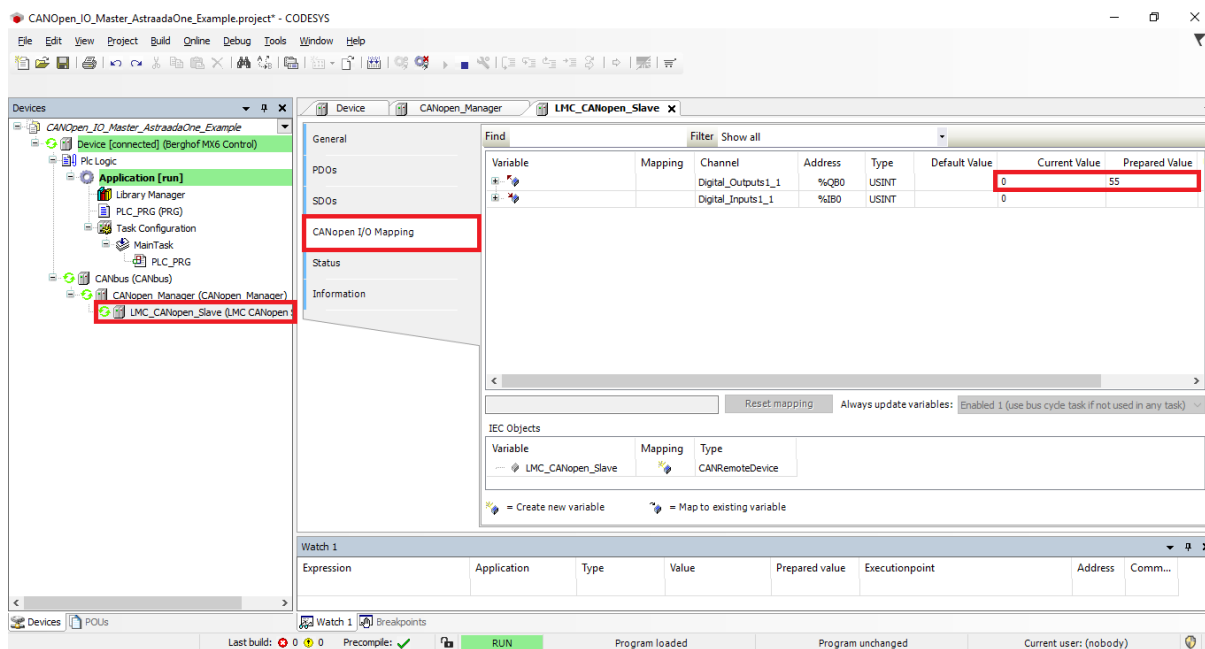
Monitorowanie wymiany danych w komunikacji CANopen

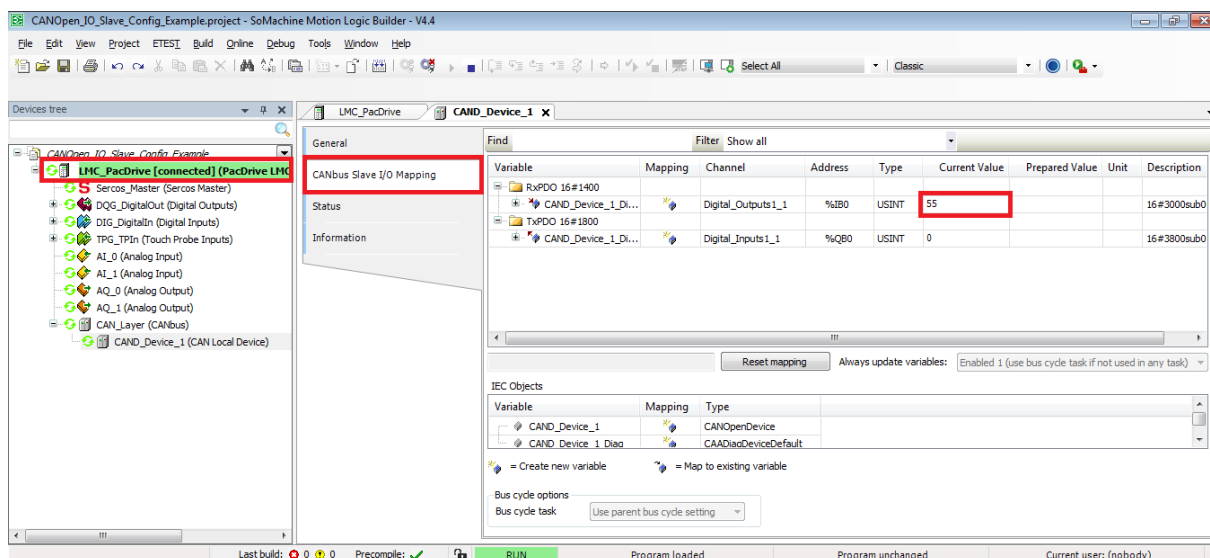
Z poziomu komputera możemy sprawdzić poprawność wymiany danych pomiędzy sterownikami wykorzystując opisane wcześniej monitorowanie zmiennych w programach SoMachine Motion Logic Builder i CODESYS.

WAŻNE! Należy pamiętać, że zmienne typu Output w urządzeniu Master są przekazywane do modułu Output w urządzeniu Slave. Analogicznie zmienne typu Input w CANopen Master odpowiadają zmiennym typu Input w urządzeniu Slave.

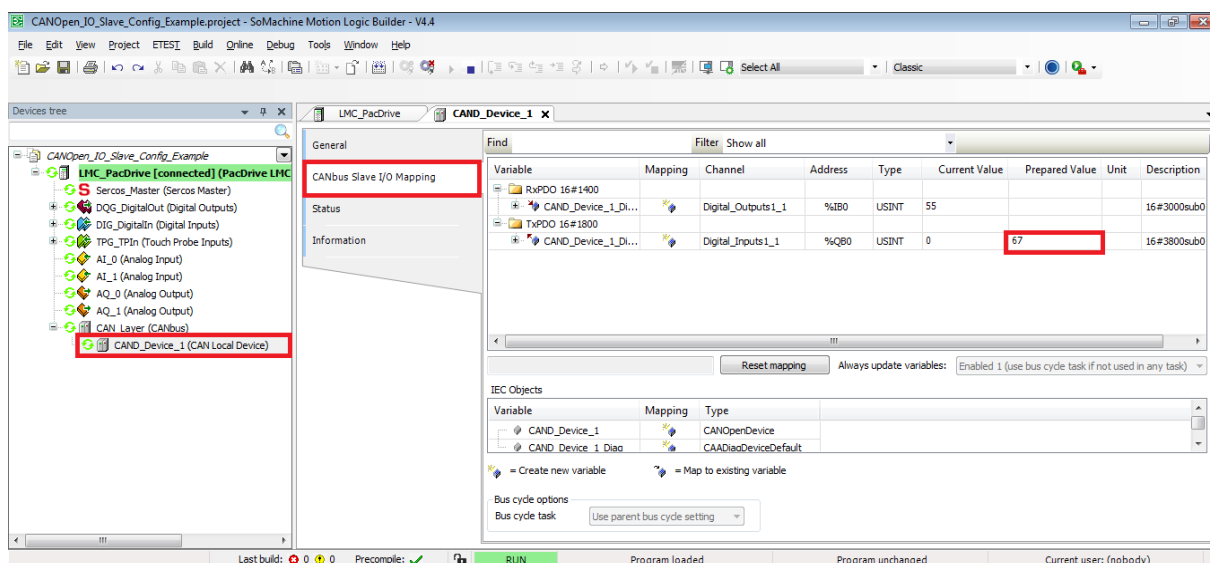
Dla konfiguracji przedstawionej w tym informatorze, z poziomu CANopen Master możemy modyfikować zmienne typu output, a odczytywać te o typie input. Z poziomu urządzenia CANopen Slave jest odwrotnie – urządzenie modyfikuje zmienne input, a odczytuje zmienne output.

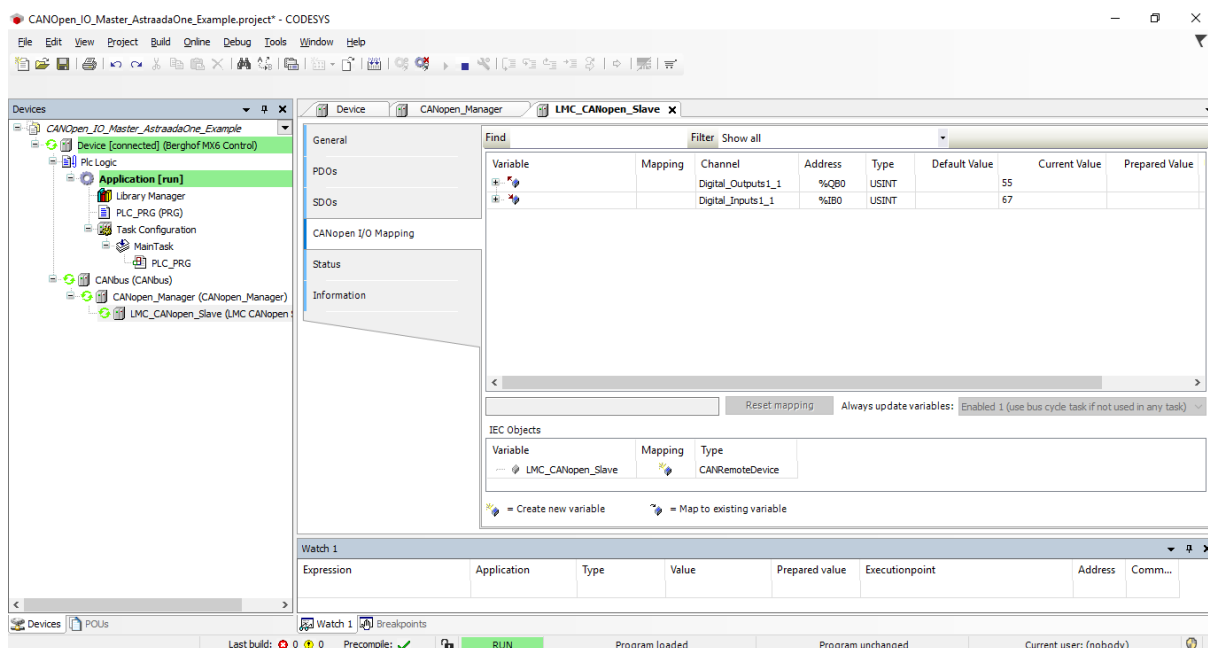
Dla przykładu w programie CODESYS, wchodząc w zakładkę *CANopen I/O Mapping* w ustawieniach *LMC_CANopen_Slave*, wpisujemy w okienku *Prepared Value* dla zmiennej *Output* wartość „55”. Wgrywamy ją do kontrolera poprzez skrót klawiszowy *Ctrl+F7*, a następnie przechodzimy do programu SoMachine Motion Logic Builder.





Jak widzimy na powyższym obrazku, CANopen Slave prawidłowo odczytuje wartość przesłaną z urządzenia Master. Dalej sprawdzamy czy moduł Master odczytuje wymuszone zmienne wejściowe nadane przez Slave. Dla przykładu w oknie *Prepared Value* dla zmiennych Input przypisano wartość „67”.





Wracając do programu *CODESYS* w zakładce *CANopen I/O Mapping* widzimy, iż wartość zmiennej Input zmieniła swoją wartość na tę nadaną z poziomu Slave. Komunikacja działa poprawnie.