

# Program Safety w oprogramowaniu CodeSys

## TWORZENIE PROGRAMU I WGRYWANIE GO DO STEROWNIKÓW SAFETY

### OPROGRAMOWANIE CODESYS

---

Przed rozpoczęciem pracy z programem Safety należy sprawdzić wersję oprogramowanie CodeSys. Funkcjonalność Safety będzie dostępna w wersjach 3.5 SP14 i wyższych. Należy zatem zaktualizować oprogramowanie pobierając instalkę ze strony Astora.

Dodatkowo oprogramowanie wymaga specjalnych plików sprzętowych Device Repository oraz Package Manager, które można uzyskać kontaktując się z przedstawicielem firmy Astor.

### PRZYGOTOWANIE CODESYS ORAZ STEROWNIKA DO PROGRAMOWANIA SAFETY

---

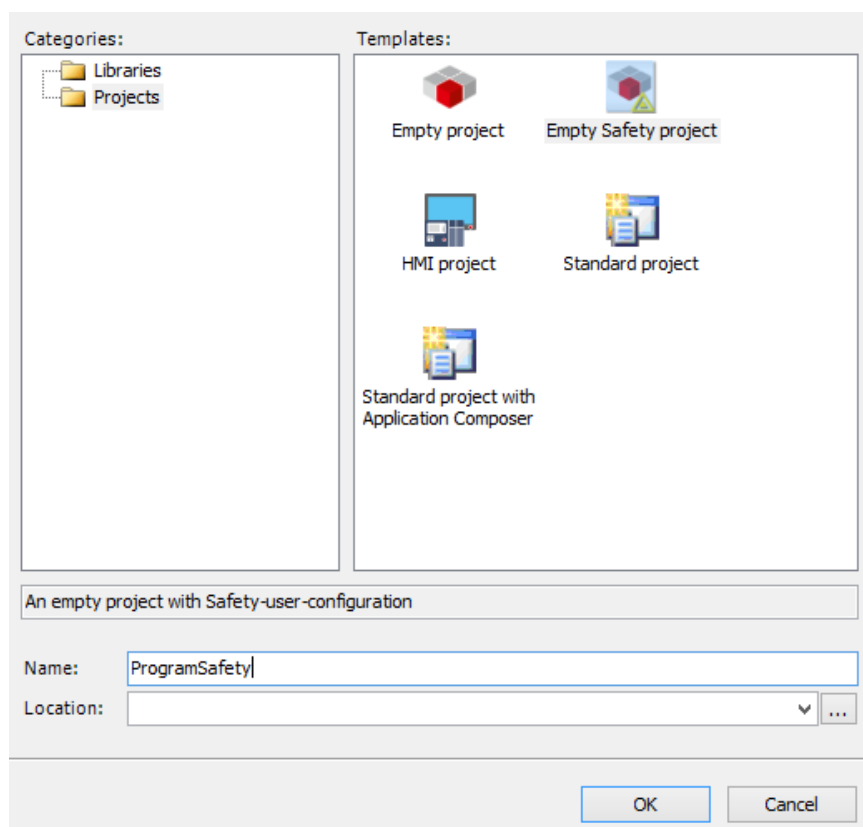
1. Sprawdź czy Codesys jest w wersji 3.5 SP14 (Certyfikowana wersja Codesys dla Safety)
2. Sprawdź czy Codesys ma doinstalowane rozszerzenie Safety (E-IO\_Safety\_Extension\_V1.5.0.0\_for\_CODESYS\_V3.5.14.40) w narzędziu Package Manager
3. Sprawdź czy sterownik ma firmware w wersji **1.22.3**
4. Sprawdź czy target w projekcie ma wersję **1.22.3**
5. Sprawdź czy zainstalowałeś odpowiednie pliki dotyczące modułów Safety za pomocą narzędzia Device Repository

## TWORZENIE PROGRAMU

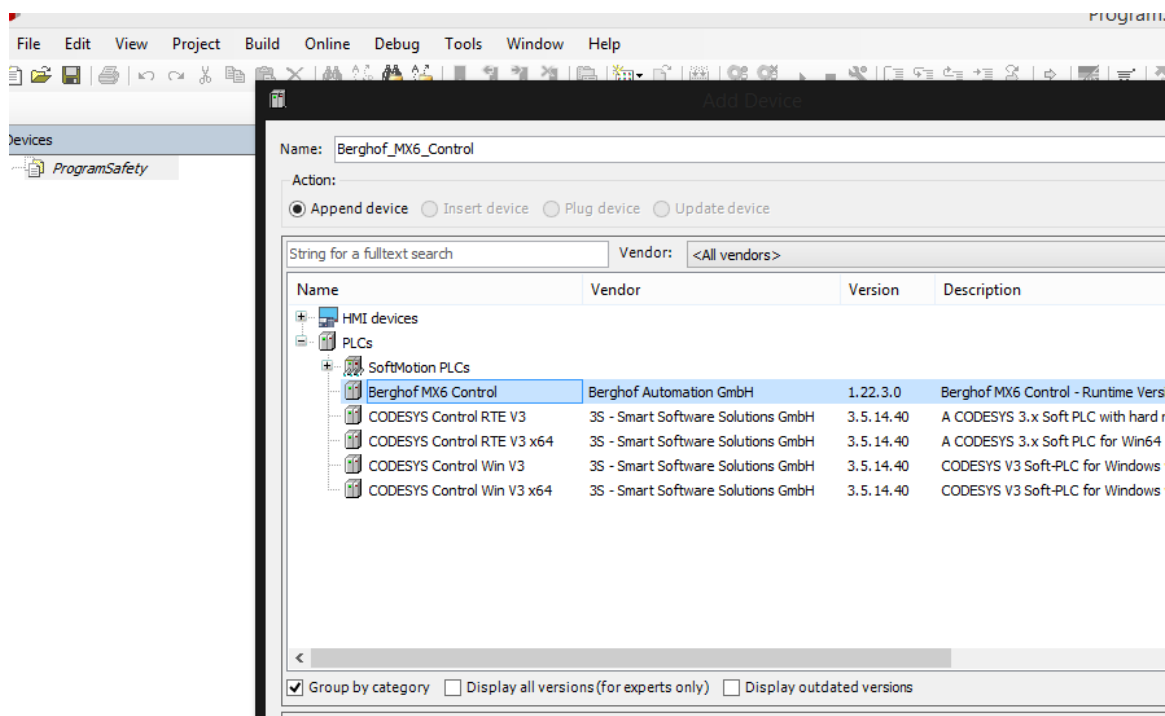
Jeżeli wersja programu CodeSys jest aktualna, oraz pliki sprzętowe zostały wgrane, można rozpocząć pracę z tworzeniem programu obsługującego funkcjonalność Safety.

Sterownik Safety pracuje jako podrzędny sterownik w sieci EtherCat. W związku z tym wymagane jest, aby współpracował z nim sterownik nadrzędny. Funkcjonalność ta znacząco uwydatnia się w momencie tworzenia drzewa projektowego programu. W związku z tym należy połączyć standardowy sterownik PLC ze sterownikiem safety. W przypadku korzystania ze sterownika kompaktowego – za pośrednictwem modułu Buscoupler, natomiast w przypadku korzystania ze sterownika modułowego – bezpośrednio przyłączenie do bocznego portu EtherCat.

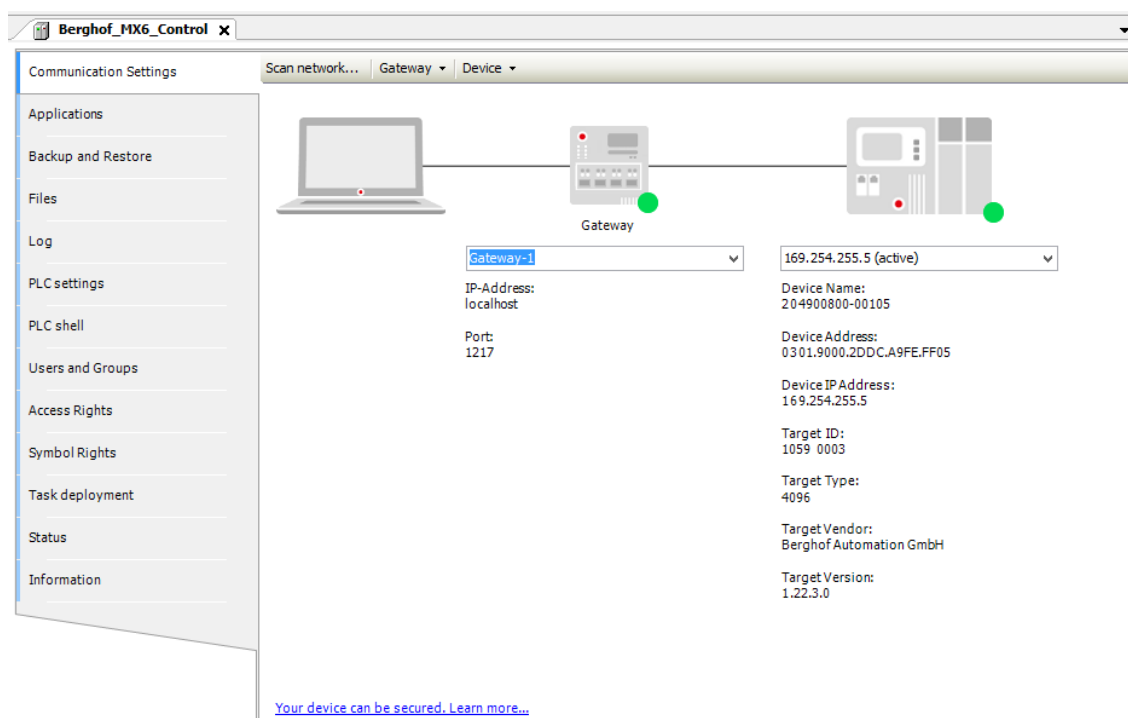
Aby rozpocząć tworzenie programu należy otworzyć oprogramowanie CodeSys, wybrać opcję *New Project*, a następnie *Empty Safety Project*, nazwać swój projekt i kliknąć *OK*.



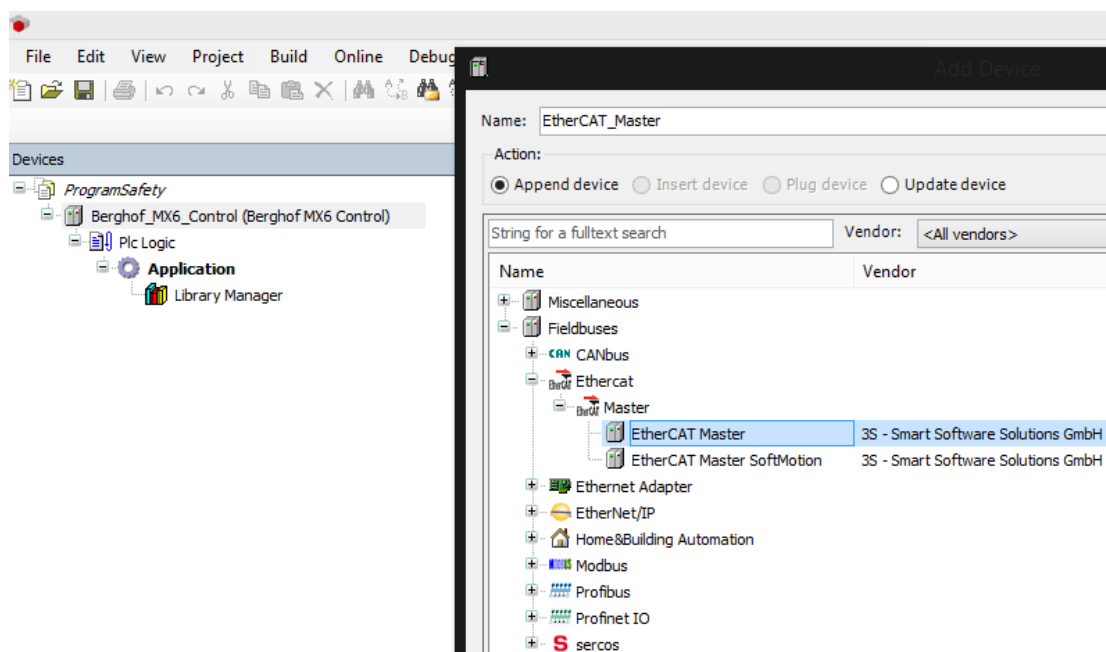
Następnie klikamy prawym przyciskiem myszy na jedyny utworzony segment w drzewku projektowym i wybieramy *Add Device*. Z listy należy wybrać sterownik nadrzędny, który jest podłączony do sterownika Safety.



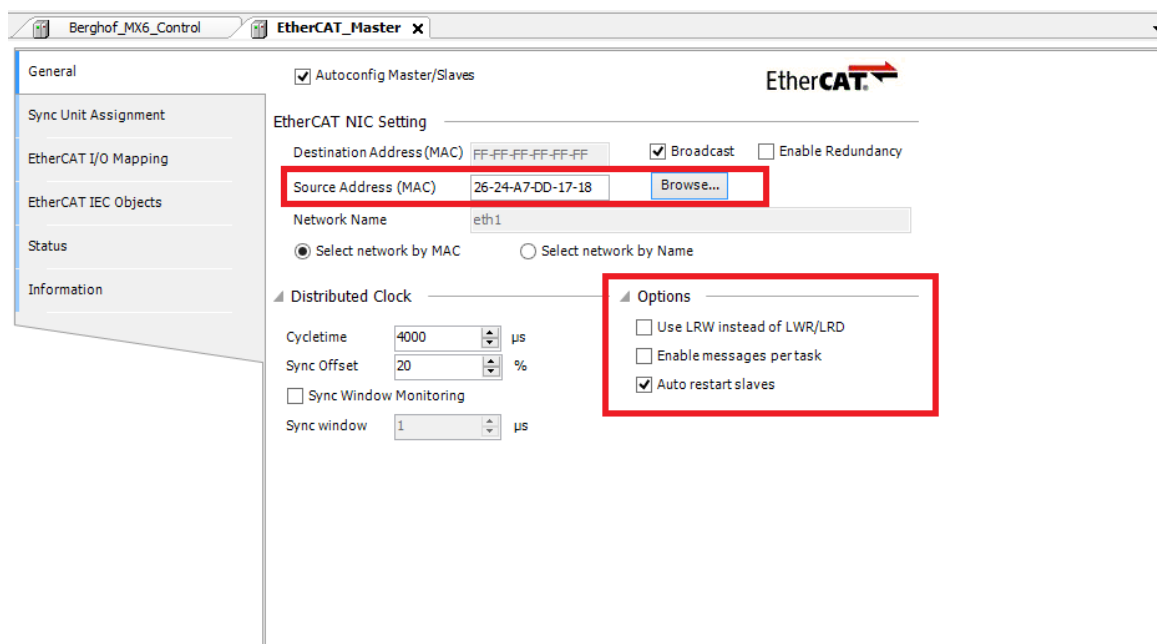
Po dodaniu sterownika należy kliknąć na niego dwukrotnie. W wyświetlonym menu *Communication Settings* podajemy adres IP i klikamy ENTER aby połączyć się ze sterownikiem.



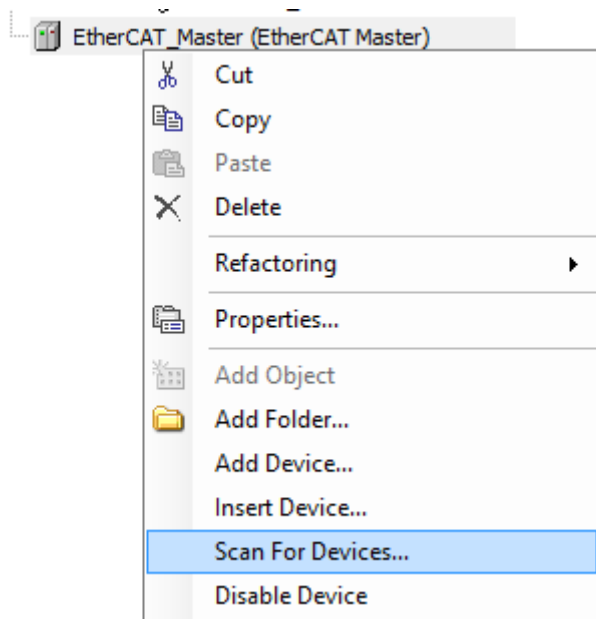
Kolejnym krokiem jest kliknięcie prawym przyciskiem myszy na dodany sterownik i wybranie ponownie opcji *Add Device* i dodanie *EtherCAT Master* do drzewa projektowego.



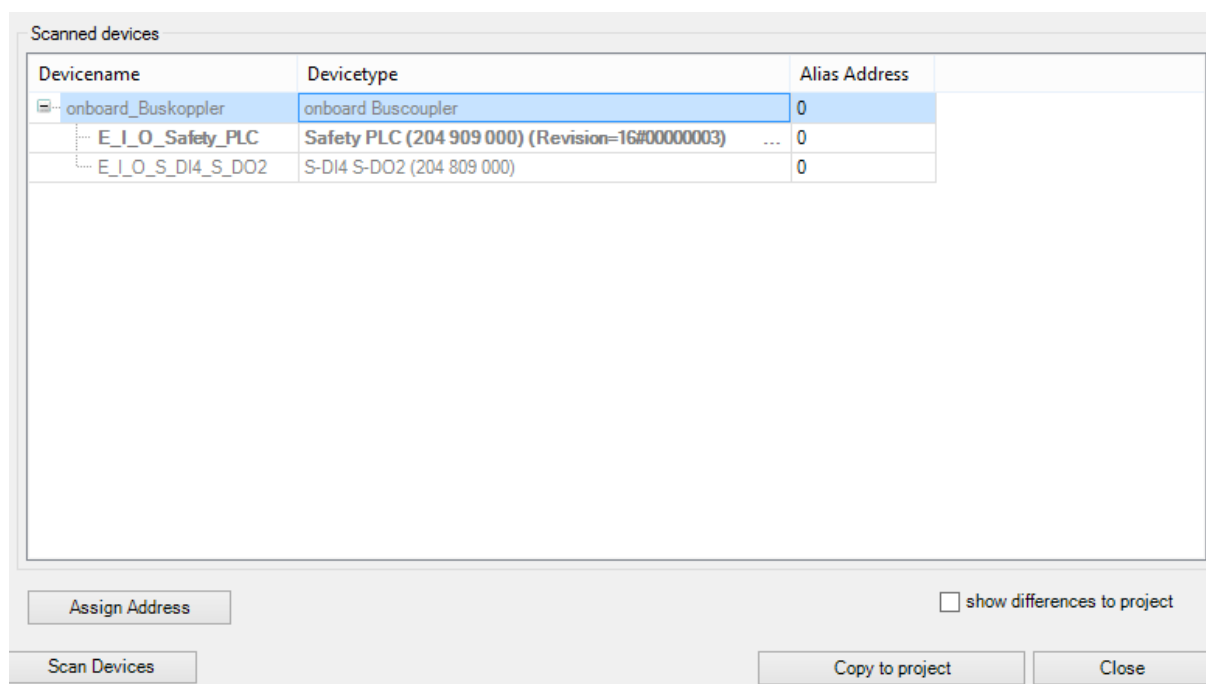
Następnie należy skonfigurować opcje mastera EtherCAT, klikając dwukrotnie na dodany moduł *EtherCAT Master*. W wyświetlonym menu *General* należy ustawić adres MAC wykorzystywanego portu (w przykładowym sterowniku jest to eth1). Warto także rozwinąć segment *Options* i zaznaczyć opcję *Auto restart slaves* – umożliwi to automatyczne nawiązywanie komunikacji z urządzeniami na wypadek utraty komunikacji.



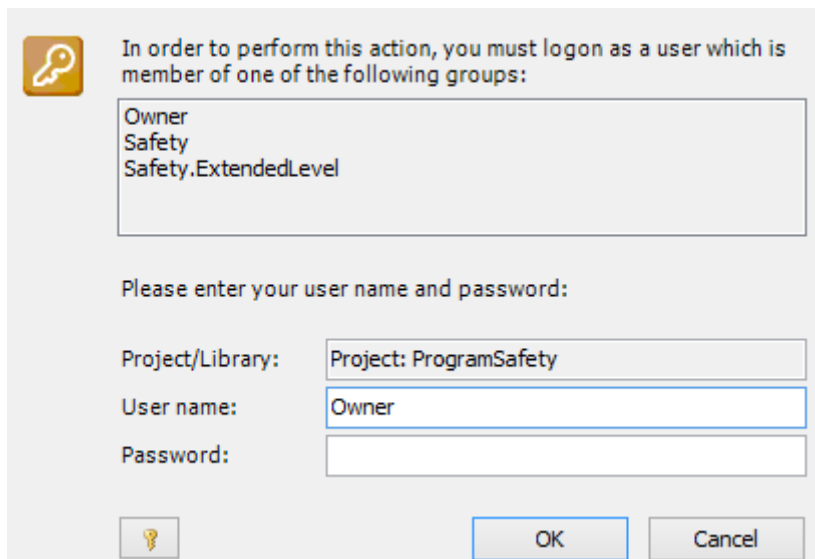
Wykonane wcześniej połączenie ze sterownikiem i skonfigurowanie opcji mastera sieci EtherCAT umożliwi wykorzystanie opcji *Scan For Devices*, która automatycznie wyszuka urządzenia podłączone do sieci EtherCat. Aby skorzystać z tej opcji należy prawym przyciskiem myszy kliknąć na *EtherCat\_Master*.



Po krótkiej chwili wyświetlą się podłączone urządzenia. W tym momencie należy zaznaczyć urządzenia, które mają znaleźć się w projekcie przytrzymując lewy klawisz ctrl i klikając lewym przyciskiem myszy. Następnie, po zaznaczeniu, należy wybrać opcję *Copy to project*.



Aplikacja poprosi o podanie loginu i hasła. Najwyższy poziom dostępu oferuje login Owner, który standardowo nie ma zaprogramowanego żadnego hasła. W związku z tym należy podać sam login i potwierdzić klawiszem OK.



In order to perform this action, you must logon as a user which is member of one of the following groups:

- Owner
- Safety
- Safety.ExtendedLevel

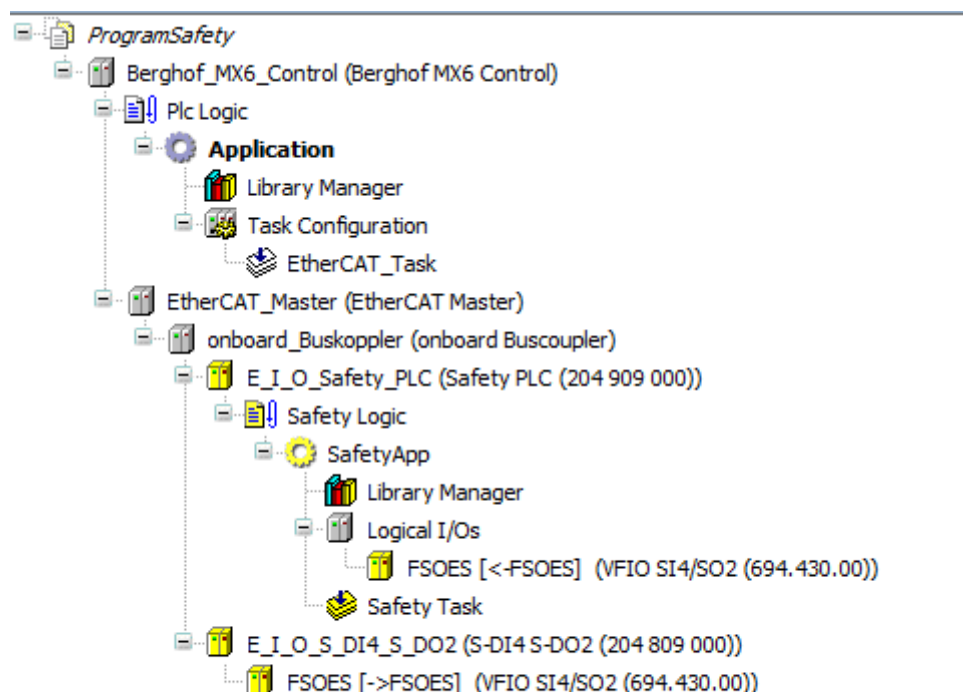
Please enter your user name and password:

Project/Library:

User name:

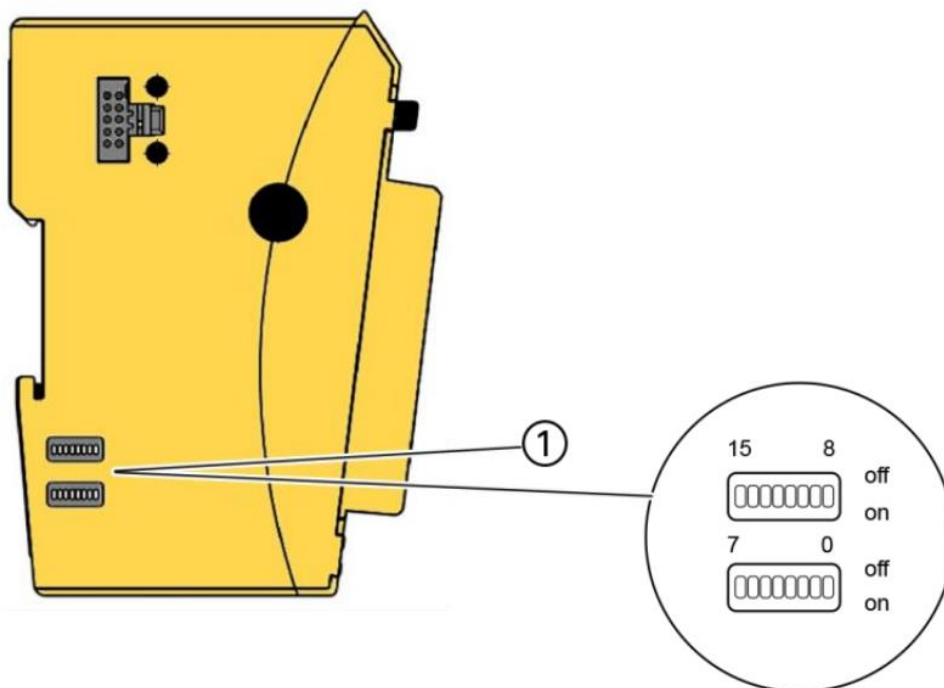
Password:

Drzewko zostanie automatycznie rozbudowane o urządzenia Safety wraz z bibliotekami i opcjami umożliwiającymi stworzenie programu Safety i wgrania go do urządzenia.

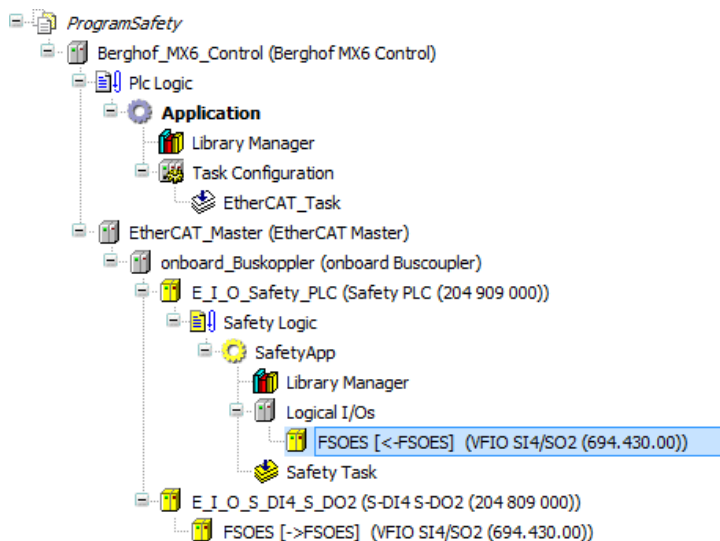


W przypadku korzystania z dodatkowych modułów I/O safety należy upewnić się czy mają one poprawnie zdefiniowany adres ID sieci FSoE. Aby to zrobić należy sprawdzić stan przełączników DIP

po wewnętrznej stronie modułu I/O. Fabrycznie wszystkie przełączniki powinny być ustawione na OFF dlatego też, należy wybrać odpowiedni adres FSoE. W tym przykładzie moduł I/O będzie miał adres 1, dlatego też należy przestawić w tryb ON drugi od prawej przełącznik z dolnego segmentu.



Po ustawieniu przełączników należy przejść do drzewa projektowego i kliknąć dwukrotnie na dodany moduł I/O w segmencie *Logical I/Os*.

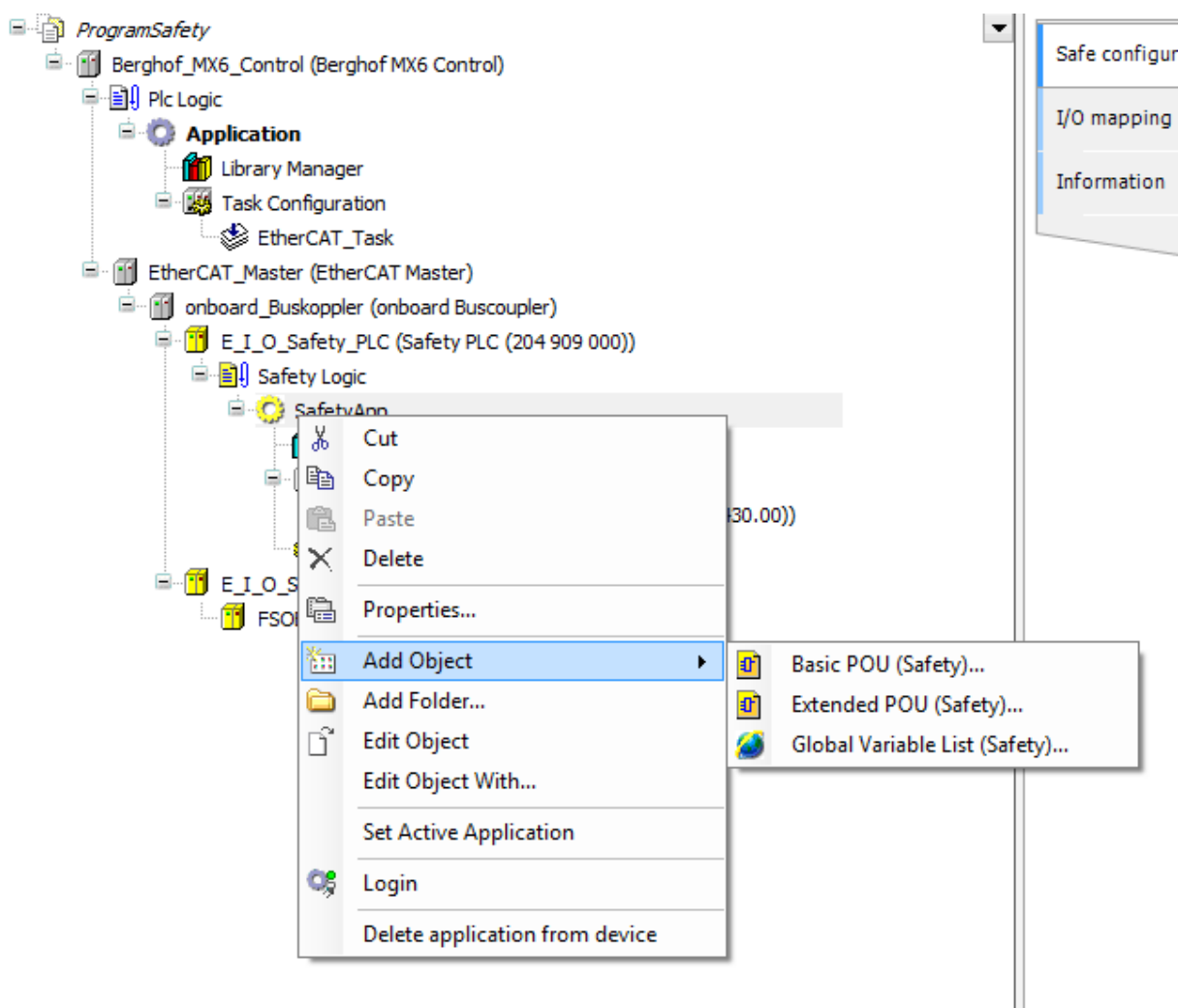


Wyświetli się menu konfiguracyjne modułu I/O, w którym należy koniecznie upewnić się czy Slave adres zgadza się z fizycznym stanem przełączników.

Safe configuration		In Work			
I/O mapping		Name	Value	Symbolic value	Description
Information		FSoE address	1		Unique FSoE slave address (dip switch setting)
		Connection ID	1		Unique id for the connection to a FSoE slave
		WatchdogTime	100		ms
		Used Inputs	15	Input 0, 1, 2, 3 used	
		External Inputs	0		
		Used Outputs	3	Output 0, 1 used	
		extGroundOutputs	0	No Outputs externa...	

W przypadku niezgodności, kompilator wskaże błąd i nie zezwoli na wgranie programu.

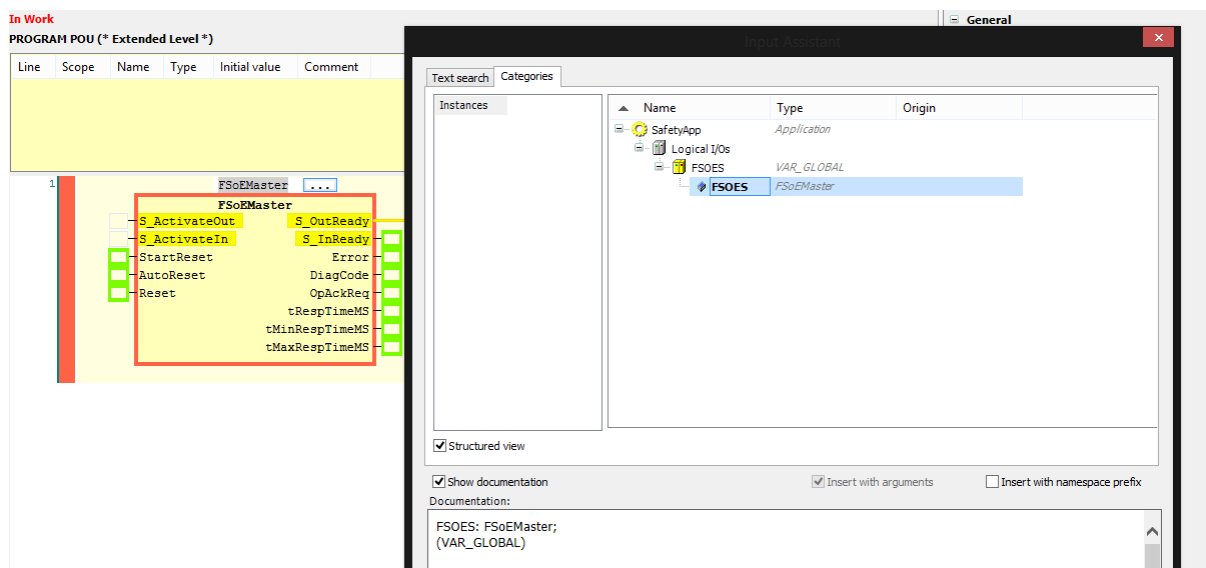
Kolejnym krokiem jest dodanie obiektu POU, w którym można zapisać program w języku Safety FBD. Aby tego dokonać należy kliknąć prawym przyciskiem myszy na *SafetyApp* i rozwinąć opcję *Add Object*. Do wyboru są dwa rodzaje obiektów POU – *Basic* oraz *Extended*. Obiekt *Basic* posiada mniej funkcjonalności, które obsługuje oraz mniej rodzajów zmiennych niż obiekt *Extended*. Dodatkowo te dwa obiekty można rozdzielić dostępem tak, aby zalogowany użytkownik miał dostęp tylko do określonego ich typu. W tym przykładzie zostanie wybrany obiekt *Extended*.





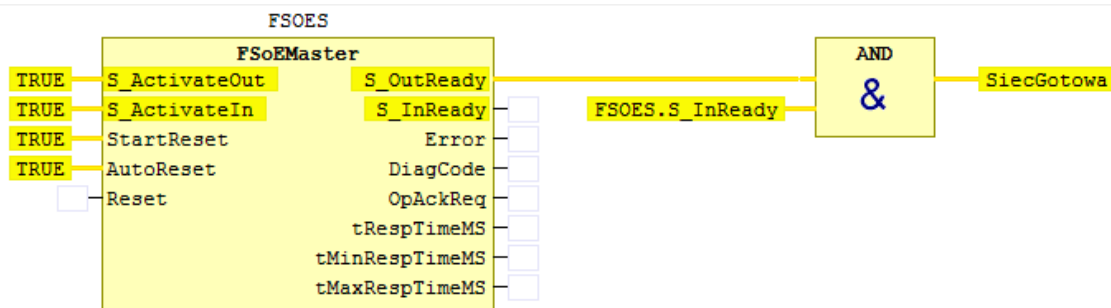
Tworzenie programu safety należy koniecznie rozpocząć od bloczka FSoEMaster. Bloczek ten sprawdza status połączenia FSoE, jak i weryfikuje diagnostykę wejść/wyjść i zezwala, bądź nie, na dalszą pracę programu. Jeżeli bloczek ten nie jest dostępny w Toolboxie, można przeciągnąć do programu dowolny bloczek, kliknąć dwukrotnie na jego nazwę i wpisać ręcznie *FSoEMaster*. Po zatwierdzeniu ENTERem, bloczek zmieni formę.

W bloczku FSoEMaster bardzo ważna jest zmienna, która ten bloczek definiuje. Nie może to być dowolna stworzona zmienna, a specjalnie przygotowana do tego zmienna, która jest sprzężona z modułem I/O. Aby skonfigurować odpowiednią zmienną należy kliknąć na klawisz trzech kropek nad bloczkiem i wybrać *SafeApp -> Logical I/Os -> FSOES -> FSOES*. Dzięki temu bloczek ten będzie sprzężony z podłączonym modulem.

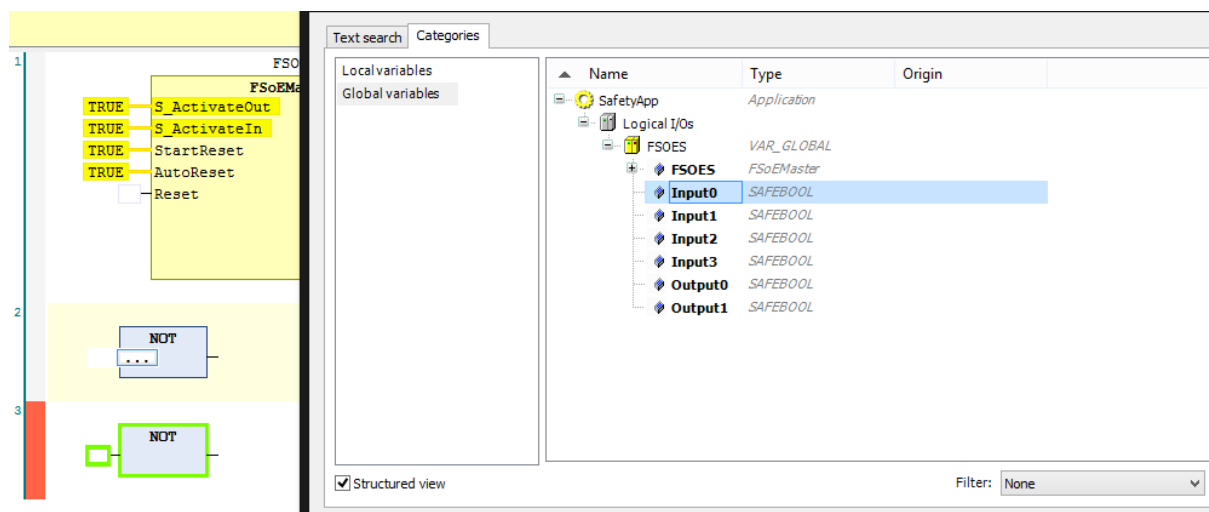


Bloczki Safety wymagają do poprawnego działania zmiennej, która będzie zezwalała na ich pracę. Zmienną tą podłącza się do wejście *Active*, które posiada każdy bloczek funkcyjny Safety (z wyjątkiem FSoEMaster, gdyż ten pracuje ciągle). W związku z tym, najlepiej wykorzystać funkcjonalność diagnostyczną bloczka FSoEMaster i zezwolić na pracę pozostałym blozkom, gdy zarówno wejścia jak i wyjścia przeszły poprawnie test diagnostyczny.

Aby stworzyć taką zmienną należy wstawić wartość *True* do wejść *S\_ActiveOut* oraz *S\_ActiveIn*, co aktywuje możliwości diagnostyczne fizycznych wejść oraz wyjść, a także do wejść *StartReset* oraz *AutoReset*, które umożliwią ciągłą i automatyczną pracę bloczka. Następnie należy wykorzystać wyjścia *S\_OutReady* oraz *S\_InReady*, które informują o poprawnym działaniu I/O - *True* lub niepoprawnym - *False*. Wyjścia te najlepiej powiązać ze sobą bloczkiem *AND*, który finalnie da nam możliwość zdefiniowania zmiennej, która będzie miała stan *True*, gdy zarówno wejścia jak i wyjścia przejdą poprawnie test diagnostyczny.



Następnie można przystąpić do dalszej części tworzenia programu. W przykładzie zostanie zdefiniowany przycisk z dwoma wejściami NC, których rozwarcie aktywuje wyjście Output 1. Bezpośrednie odwołanie do fizycznych wejść oraz wyjść modułu I/O znajduje się w segmencie *Global variables*, po rozwinięciu drzewka opcji.

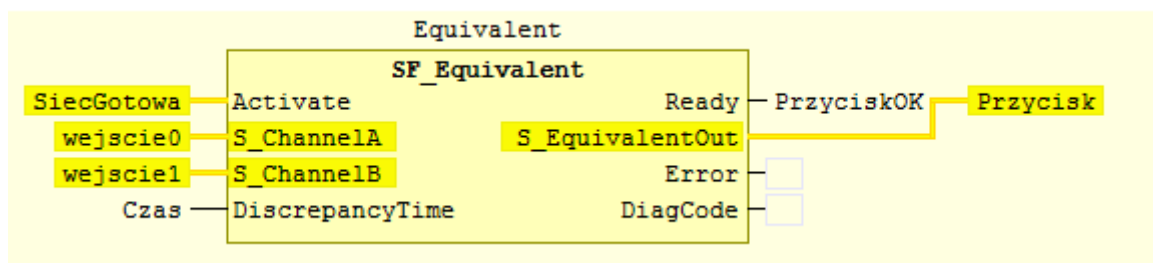


Należy przetworzyć dwa sygnały wejściowe w jeden wyjściowy. Aby tego dokonać trzeba skorzystać z bločku *SF\_Equivalent* – jeżeli styki w przycisku są tego samego rodzaju lub *SF\_Antivalent* – jeżeli są dwa różne styki. W tym przykładzie styki są tego samego rodzaju (NC) dlatego wykorzystany zostanie pierwszy bloček.

Aby skonfigurować poprawnie bloček *SF\_Equivalent* należy:

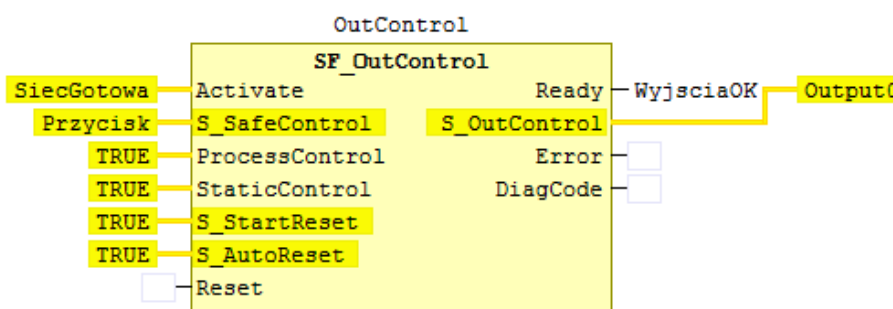
- Na wejściu *Active* podać zmienną, która aktywuje działanie bločku. W tym przykładzie jest to przygotowana wcześniej zmienna *SiecGotowa*.
- Do wejść *S\_Channel1A* oraz *S\_Channel1B* podłączyć sygnały wejściowe z przycisku.
- Stworzyć zmienną *Czas*, w której zostanie zdefiniowany maksymalny czas w jakim oba sygnały mają zmienić stan na *True*. W tym przykładzie styki działają w ramach jednego przycisku więc sygnały zmieniają się niemal jednocześnie, a więc czas ten może być krótki.

- Na wyjściach należy zdefiniować zmienną przy wyjściu *Ready*, która informuje o gotowości do działania bloczka, oraz przy wyjściu *S\_EquivalentOut*, która będzie główną zmienną wyjściową.

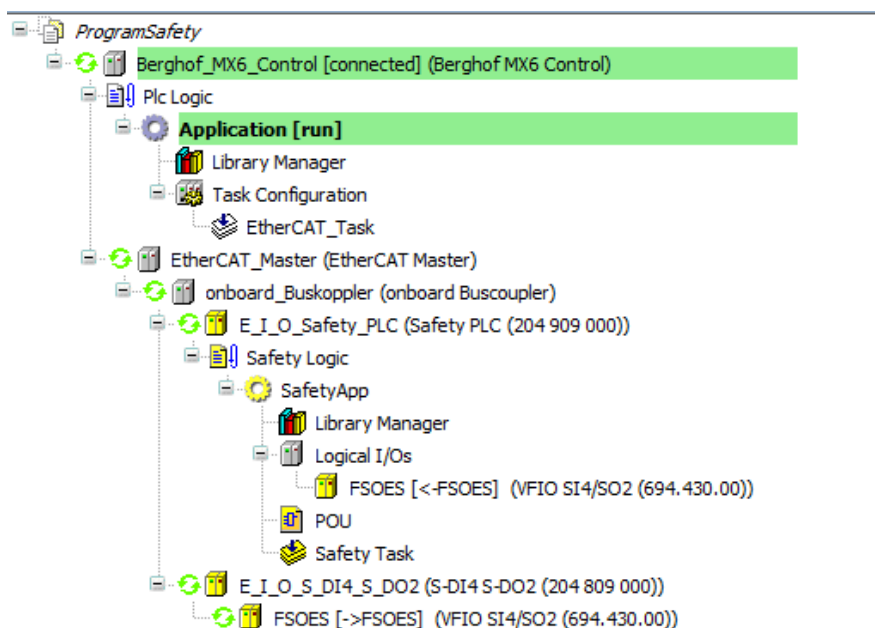


Należy pamiętać, że główną zmienną wyjściową nie może być bezpośrednio wyjście modułu I/O. Program Safety wymaga wykorzystania osobnego bloczku funkcyjnego, który obsługuje fizyczne wyjścia – *SF\_OutControl*. Aby poprawnie skonfigurować ten bloczek należy:

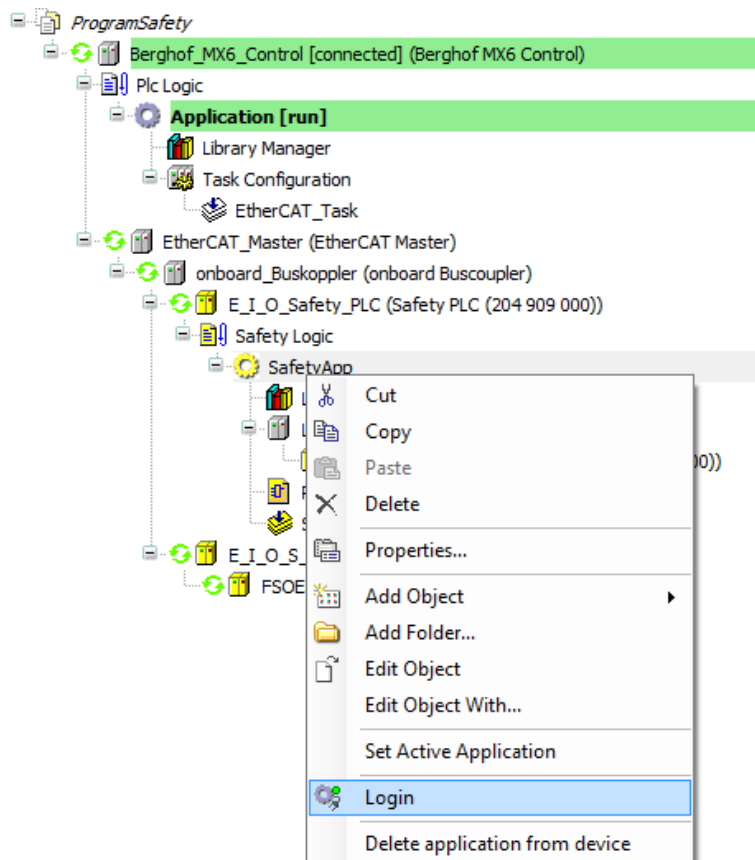
- Tak jak w poprzednim przypadku użyć zmiennej przy wejściu *Active*.
- Do wejścia *S\_SafeControl* należy podłączyć główny sygnał wejściowy z bloczka *SF\_Equivalent*. W tym przykładzie jest to zmienna *Przycisk*.
- Wstawić wartość *True* do wejść *ProcessControl* i *StaticControl*. Są to wejścia, które umożliwiają dodatkową aktywację, bądź dezaktywację działania bloczka z zewnętrznego programu, który w tym przykładzie nie jest definiowany. Wartość tą należy też wstawić do wejść *S\_StartReset* oraz *S\_AutoReset*, aby zapewnić ciągłą pracę bloczka.
- Do wyjścia *Ready* należy podłączyć zmienną, która będzie informować o gotowości do działania bloczka, natomiast do wyjścia *S\_OutControl* należy podłączyć bezpośrednio fizyczne wyjście *Output0* modułu I/O.



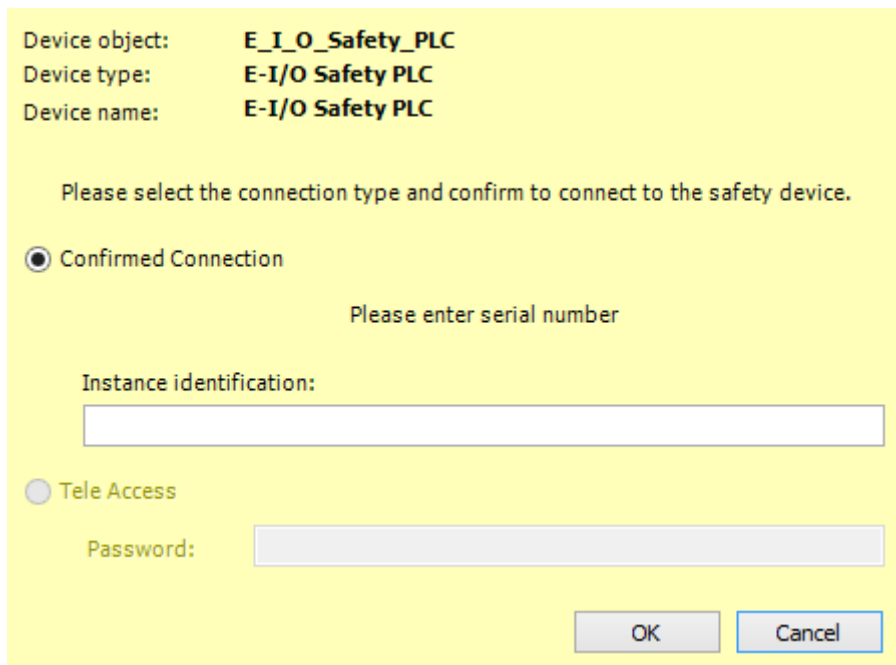
Następnie należy przystąpić do wgrывania programu. Pierwszym krokiem jest nawiązanie komunikacji z głównym sterownikiem i zalogowanie się do niego wraz z wgraniem programu. Poprawna komunikacja będzie sygnalizowana zielonymi strzałkami po wprowadzeniu sterownika w tryb RUN.



Kolejnym krokiem jest zalogowanie się bezpośrednio do sterownika Safety. Aby to zrobić należy kliknąć prawym przyciskiem myszy na *SafetyApp* i wybrać opcje *Login*.



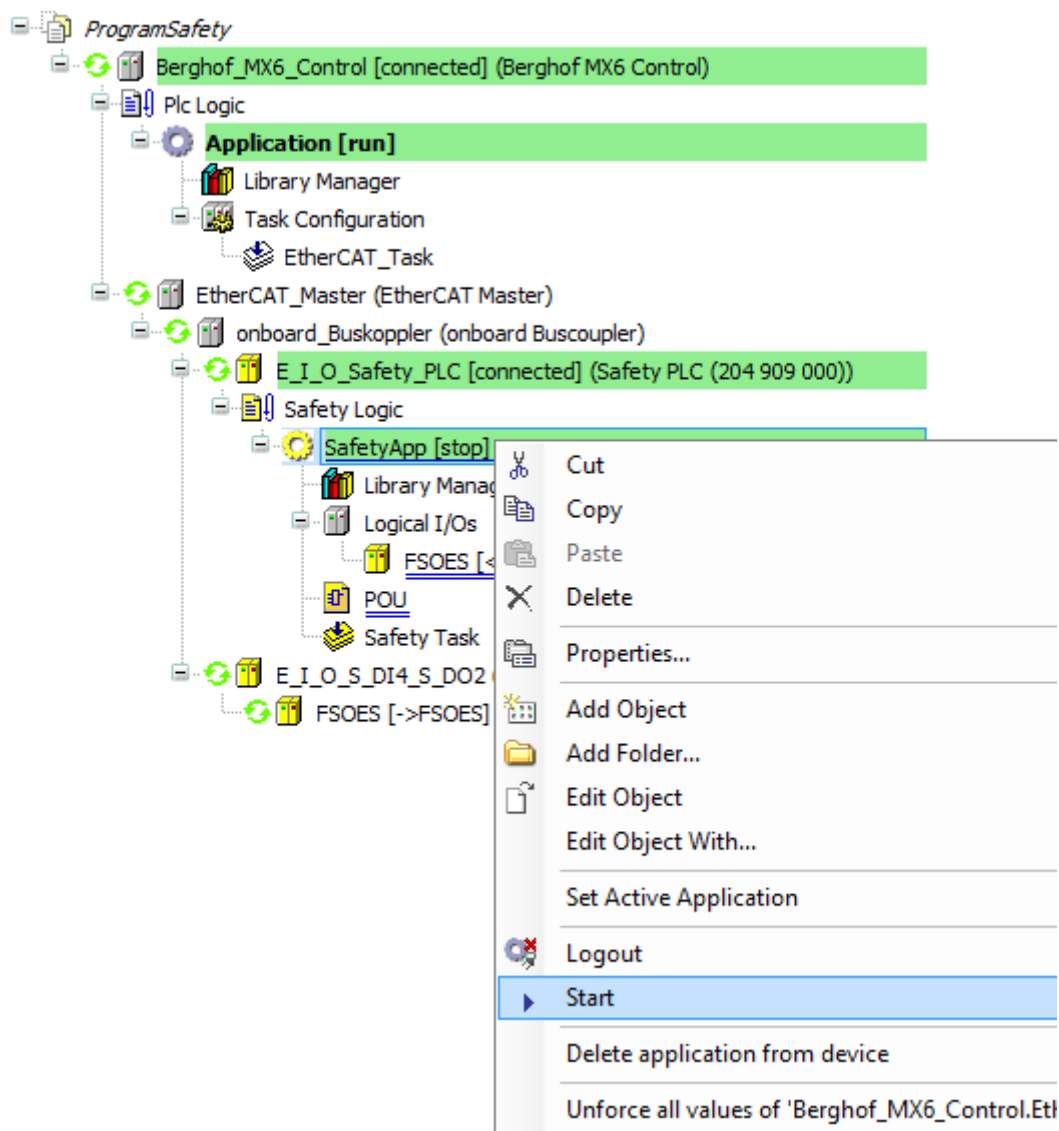
Aplikacja poprosi o weryfikację logowania poprzez podanie numeru seryjnego sterownika Safety. Należy taki numer podać i wcisnąć *OK*.



Następnie pojawi się komunikat informujący o tym, że aplikacja różni się od tej aktualnie wgranej do sterownika. Potwierdzenie tego komunikatu przełączy sterownik w tryb testowy umożliwiający wgranie nowej aplikacji i przetestowanie go fizycznie na sprzęcie.

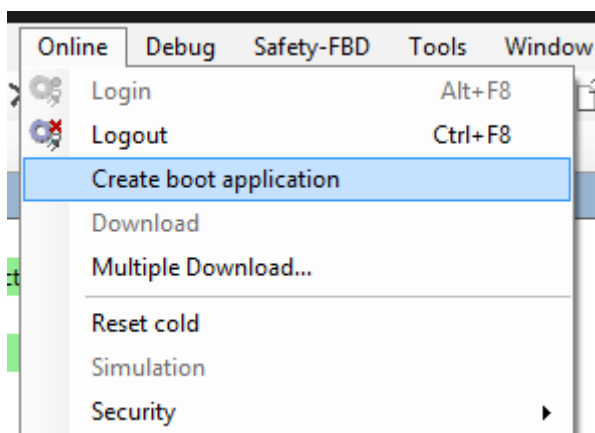
Ostatnim komunikatem jest prośba o podanie hasła. Fabrycznie hasło nie jest zaprogramowane, więc wystarczy potwierdzić ten komunikat klawiszem *OK* bez wypełniania pola z hasłem.

Po wgraniu aplikacji należy przełączyć sterownik Safety w tryb RUN klikając prawym przyciskiem myszy na *SafetyApp* i wybierając opcję *Start*. W tym momencie należy przeprowadzić test funkcjonalności aplikacji.



W ten sposób aplikacja została wgrana do sterownika Safety, jednak po wylogowaniu zostanie przywrócona aplikacja sprzed logowania. W związku z tym, jeżeli test potwierdził poprawność działania aplikacji należy przygotować plik Boot. Aby tego dokonać, należy najpierw kliknąć prawym przyciskiem myszy na *SafetyApp* i wybrać opcję *Set Active Application*.

Następnie należy przejść do lewego górnego rogu ekranu, rozwinąć opcję *Online* i wybrać *Create boot application*.



Po wylogowaniu ze sterownika, oprogramowanie zapyta czy sterownik ma zostać zrestartowany i uruchomiony z aktualną wersją aplikacji boot, którą ma wgrać do pamięci. Komunikat ten należy potwierdzić.

Od tego momentu sterownik będzie realizował napisaną aplikację.