

# BECKHOFF

## TwinCAT 3 - FAQ

**Zbiór odpowiedzi na najczęściej zadawane pytania  
związane z oprogramowaniem TwinCAT 3**

Wersja dokumentacji 1.0

Aktualizacja: 08.04.2020

Kontakt: *support@beckhoff.pl*

Beckhoff Automation Sp. z o. o.

## Spis treści

1	Wstęp.....	6
2	Środowisko TwinCAT 3 – informacje podstawowe .....	7
2.1	Instalacja .....	7
2.2	Przełączanie między środowiskami .....	7
2.3	Uruchamianie TwinCAT 3 na lokalnym komputerze .....	9
2.3.1	Możliwie występujące problemy podczas uruchamiania .....	9
2.3.2	Włączanie obsługi wirtualizacji w BIOS.....	9
2.3.3	Izolacja rdzenia w procesorze .....	9
2.3.4	Przywracanie wyizolowanego rdzenia.....	10
2.3.5	Plik win8settack.bat.....	12
2.4	Projekt bootowalny, kod źródłowy na sterowniku.....	12
2.4.1	Wgrywanie projektu bootowalnego .....	12
2.4.2	Wgrywanie kodu źródłowego .....	12
2.4.3	Pobieranie kodu źródłowego ze sterownika .....	13
2.5	Offline / online Beckhoff Information System.....	14
3	Program PLC i programowanie .....	16
3.1	Eksport poszczególnych części składowych projektu .....	16
3.1.1	Eksport pliku .plcproj .....	16
3.1.2	Import poszczególnych plików składowych projektu .....	16
3.1.3	Eksport do pliku .zip.....	17
3.1.4	Import z pliku .zip .....	17
3.1.5	Eksport w formacie .tnzip .....	18
3.2	Definiowanie własnych skrótów klawiszowych .....	18
3.3	Dostęp do zmiennych .....	19
3.3.1	Zmienne globalne .....	19
3.3.2	Zmienne lokalne .....	20
3.4	Wywoływanie bloków funkcyjnych .....	21
3.4.1	Wywołanie bloku funkcyjnego w języku ST .....	21
3.4.2	Przekazywanie wartości z bloku funkcyjnego.....	22
3.5	Debuggowanie .....	23
3.5.1	Sprawdzanie wszystkich składowych projektu .....	23

3.5.2	Checks.....	23
3.5.3	Breakpoints.....	24
3.5.4	Znajdowanie referencji w projekcie .....	26
3.5.5	Sprawdzanie obciążenia sterownika.....	26
3.5.6	Sprawdzanie czasu wykonywania tasku .....	26
3.5.7	Dodatkowe opcje debuggowania .....	27
3.6	Find & replace.....	28
3.7	Komentarze.....	29
3.7.1	Komentarze w języku ST .....	29
3.7.2	Komentarze w językach IL, LD, FBD .....	29
3.8	Atrybuty .....	30
3.8.1	TcContextName .....	30
3.8.2	TcLinkTo .....	30
3.8.3	Displaymode .....	31
3.8.4	Hide.....	31
3.8.5	Hide_all_locals.....	31
3.8.6	Strict.....	32
3.9	Zmiana ustawień środowiska .....	32
3.9.1	Zmiana czcionki środowiska .....	32
3.9.2	Zmiana stylu środowiska .....	33
3.9.3	Zmiana domyślnego koloru zaznaczania .....	33
3.9.4	Zmiana długości wyświetlanych zmiennych w podglądzie online.....	34
4	Biblioteki .....	35
4.1	Zapisywanie projektu w formie biblioteki .....	35
4.2	Dodawanie bibliotek systemowych do projektu .....	35
4.3	Korzystanie ze specyficznej wersji biblioteki .....	36
4.4	Instalowanie własnych bibliotek.....	37
4.5	Przekazywanie projektów z bibliotekami własnymi .....	37
4.6	Kopiowanie bibliotek z TC2 do TC3.....	38
5	Nowe typy danych w TwinCAT 3 .....	39
5.1	Union .....	39
5.2	BIT .....	40

5.3	LTIME .....	40
5.4	REFERENCE.....	41
5.5	Zmiana typu wyliczeniowego enumeracji .....	41
6	Tips & tricks .....	42
6.1	Watchlist.....	42
6.2	Otwieranie aplikacji stworzonej w TwinCAT 2 w TwinCAT 3.....	42
6.3	Dodawanie nowych tasków .....	43

© Beckhoff Automation Sp. z o.o.

*Wszystkie obrazy są chronione prawem autorskim. Wykorzystywanie i przekazywanie osobom trzecim jest niedozwolone.*

*Beckhoff®, TwinCAT®, EtherCAT®, Safety over EtherCAT®, TwinSAFE®, XFC® i XTS® są zastrzeżonymi znakami towarowymi i licencjonowanymi przez Beckhoff Automation GmbH. Inne oznaczenia użyte w niniejszej prezentacji mogą być znakami towarowymi, których użycie przez osoby trzecie do własnych celów może naruszać prawa właścicieli.*

*Informacje przedstawione w tej prezentacji zawierają jedynie ogólne opisy lub cechy wydajności, które w przypadku rzeczywistego zastosowania nie zawsze mają zastosowanie zgodnie z opisem lub które mogą ulec zmianie w wyniku dalszego rozwoju produktów. Obowiązek przedstawienia odpowiednich cech istnieje tylko wtedy, gdy zostanie to wyraźnie uzgodnione w warunkach umowy.*

**Uwaga!** Poniższy dokument zawiera przykładowe zastosowanie produktu oraz zbiór zaleceń i dobrych praktyk. Służy on wyłącznie celom szkoleniowym i wymaga szeregu dalszych modyfikacji przed zastosowaniem w rzeczywistej aplikacji. Autor dokumentu nie ponosi żadnej odpowiedzialności za niewłaściwe wykorzystanie produktu. Dany dokument w żadnym stopniu nie zastępuje dokumentacji technicznej dostępnej online na stronie [infosys.beckhoff.com](http://infosys.beckhoff.com).



## 1 Wstęp

Niniejsza instrukcja zawiera tzw. FAQ (ang. *Frequently Asked Questions* – *Często zadawane pytania*) i jest swoistym zbiorem odpowiedzi na najczęściej pojawiające się wśród użytkowników pytania związane z obsługą oprogramowania TwinCAT 3. Powinna ona ułatwić użytkownikowi poruszanie się w oprogramowaniu a także przedstawić tzw. Tips & Tricks korzystania z TC3. **Nie jest to instrukcja podstawowej obsługi oprogramowania i sprzętu firmy Beckhoff i nie zastępuje szkolenia TwinCAT 3 PLC!**

Od użytkownika instrukcji wymaga się podstawowej wiedzy na temat programowania sterowników firmy Beckhoff w środowisku TwinCAT 3.

Do stworzenia instrukcji użyto sterownika CX9020-0115 z Windows Embedded Compact 7 oraz TwinCAT 3.1.4024.7 a także komputera z 64-bitowym systemem operacyjnym Windows 10 i oprogramowaniem TwinCAT 3.1.4024.7 oraz środowiskiem Microsoft Visual Studio 2013.

**Lista dostępnych instrukcji wraz z linkami z których można je uzyskać znajdować się będzie na końcu dokumentu!**

## 2 Środowisko TwinCAT 3 – informacje podstawowe

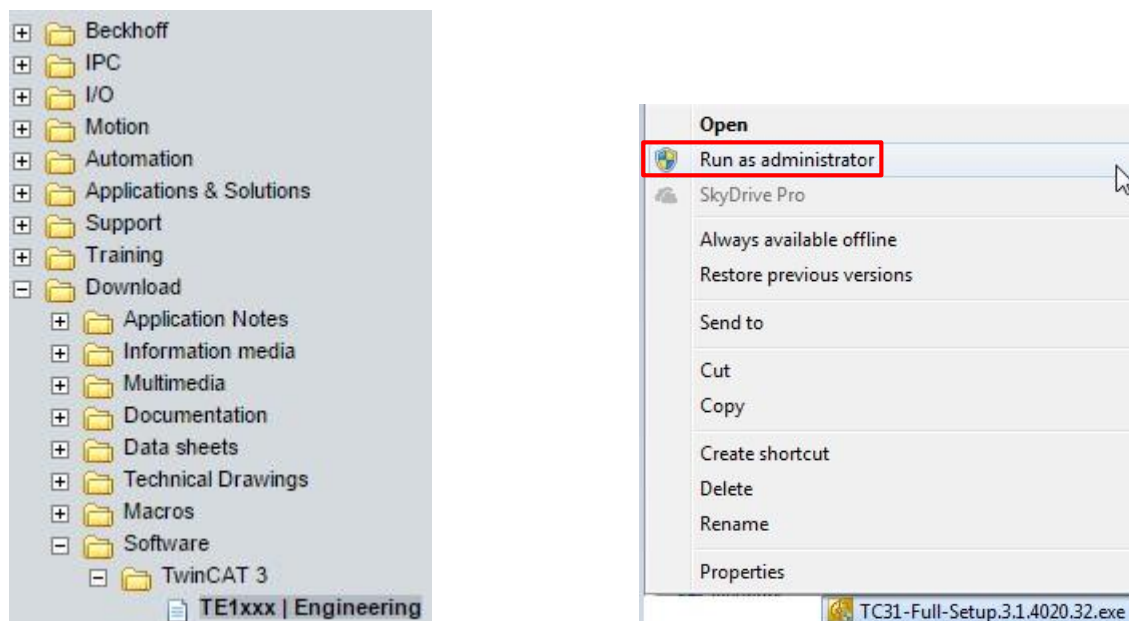
### 2.1 Instalacja

Plik instalacyjny TwinCAT 3 należy pobrać ze strony [www.beckhoff.pl](http://www.beckhoff.pl). Nie jest wymagana wcześniejsza instalacja środowiska VisualStudio ponieważ plik instalacyjny zawiera wszystkie potrzebne komponenty.

Ścieżka do pliku na stronie:

**Download -> Software > TwinCAT 3 -> TE1xxx | Engineering**

**Plik instalacyjny należy uruchamiać jako Administrator!**



Jeśli na komputerze mają znajdować się oba środowiska (TC2/TC3), to instalację zaleca się przeprowadzić w kolejności **TC2 → TC3**. Po instalacji TC3 na pasku zadań, obok zegara, powinna się pojawić ikona:



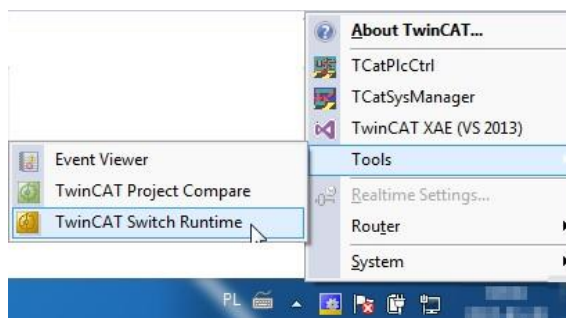
### 2.2 Przełączanie między środowiskami

**Uruchamianie runtime TwinCAT 2 działa wyłącznie na systemach 32-bitowych (architektura x86)!**

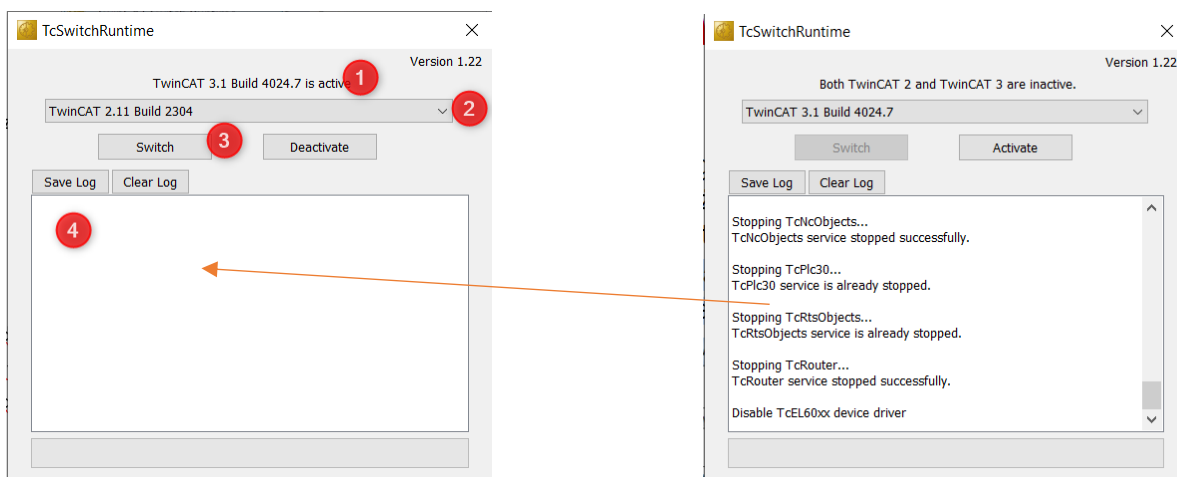
Jeśli na komputerze zainstalowane są oba środowiska, przełączanie pomiędzy nimi następuje za pomocą narzędzia **TwinCAT Switch Runtime**. Przy aktywnym środowisku TC3 narzędzie to można wywołać w następujący sposób:

Ścieżka do narzędzia na dysku:

**Dysk (C: ) -> TwinCAT -> TwinCAT Switch Runtime -> TwinCAT Switch Runtime.exe**



Po uruchomieniu narzędzie pojawi się następujące okno:



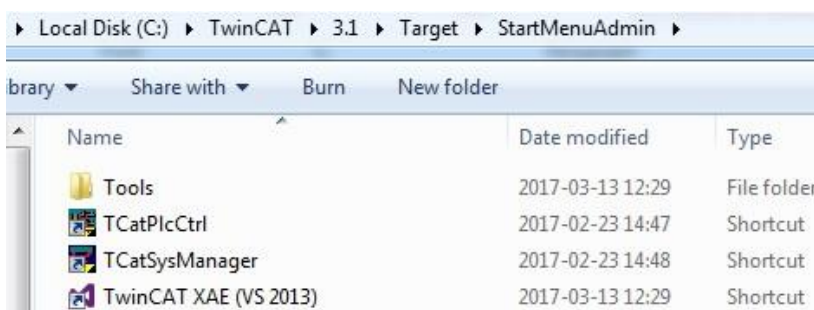
Na górze znajduje się informacja o aktywnym środowisku (1), poniżej znajduje się rozwijane menu (2), z którego należy wybrać na które środowisko chcemy się przełączyć. Po wybraniu należy kliknąć na przycisk Switch (3). Przycisk jest nieaktywny gdy wszystkie instancje TwinCAT (4) są wyłączone.

Jeśli chcemy uruchamiać oba środowiska z jednego miejsca, należy do lokalizacji:

**Dysk (C: ) -> TwinCAT -> 3.1 -> Target -> StartMenuAdmin** wkleić skróty do aplikacji TwinCAT PLC Control oraz TwinCAT System Manager. Aplikacje te znajdują się odpowiednio w lokalizacjach:

**Dysk (C: ) -> TwinCAT -> Plc -> TCatPlcCtrl.exe**

**Dysk (C: ) -> TwinCAT -> Io -> TCatSysManager.exe**





## 2.3 Uruchamianie TwinCAT 3 na lokalnym komputerze

System TwinCAT 3 pozwala na uruchomienie Runtime'u na lokalnym komputerze – komputer staje się wtedy sterownikiem PLC. Pozwala to na sprawdzenie działania algorytmu bez posiadania fizycznego IPC i wysp I/O.

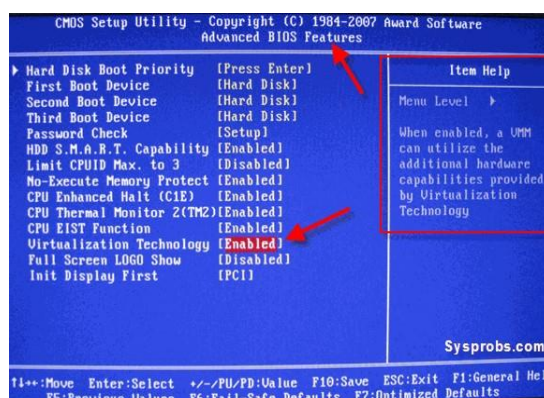
### 2.3.1 Możliwie występujące problemy podczas uruchamiania

Jeśli podczas logowania się TC3, przy urządzeniu docelowym wybranym jako <Local> (lokalny komputer), pojawia się błąd uniemożliwiający uruchomienie programu, należy wykonać jeden lub kilka z poniżej podanych kroków:

- włączyć obsługę wirtualizacji w BIOS ( na systemach x64)
- wyizolować rdzeń
- uruchomić plik win8settack.bat
- sprawdzić czy nie ma konfliktu pomiędzy środowiskiem TwinCAT a oprogramowaniem antywirusowym etc.

### 2.3.2 Włączanie obsługi wirtualizacji w BIOS

Aby aktywować wirtualizację należy aktywować (przełączyć na Enabled) poniższą opcję w BIOS ( zdjęcie poglądowe, ustawienia mogą różnić się w zależności od producenta komputera i systemu operacyjnego):

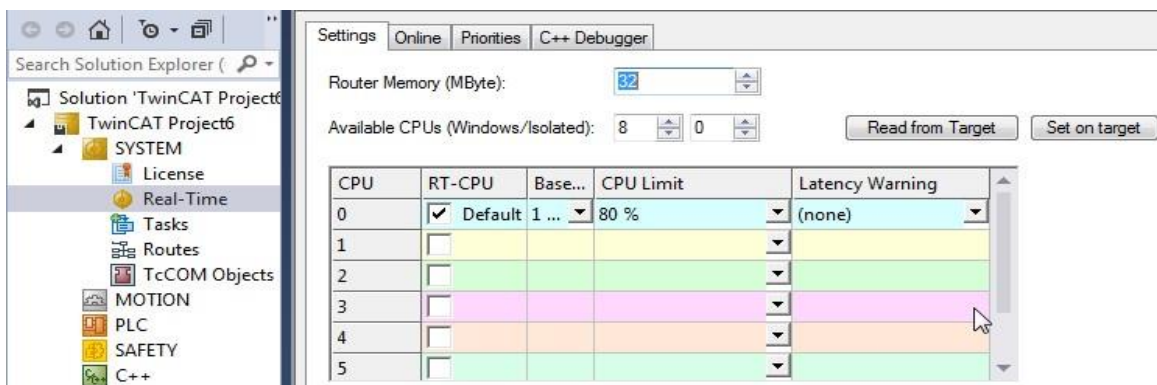


### 2.3.3 Izolacja rdzenia w procesorze

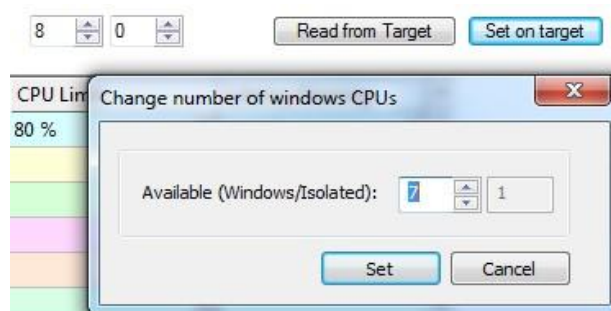
Jeśli nie możemy uruchomić TwinCAT 3 lokalnie na komputerze może istnieć konieczność wyizolowania rdzenia. Aby to zrobić najpierw należy otworzyć (lub utworzyć) dowolny projekt i jako urządzenie docelowe wybrać <Local>.



Następnie należy odnaleźć ustawienia **Real-Time** w drzewie projektu i w zakładce **Settings** kliknąć na przycisk **Read from Target** (wyświetlone dane będą różnić się w zależności od komputera).



Następnie w tej samej zakładce wybieramy przycisk **Set on Target**. W oknie **Change numbers of Windows CPU** edytujemy ustawienia w ten sposób aby w polu Isolated znalazła się wartość 1. Po zmianie wybieramy **Set** i potwierdzamy pojawiające się komunikaty. Zmiana tych ustawień wymaga restartu komputera.



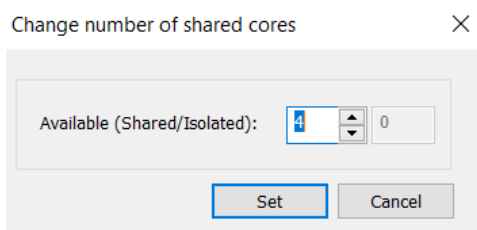
Po restarcie komputera należy wrócić do zakładki **Real-Time** -> **Settings**. Na liście dostępnych wątków, jeden pojawi się z atrybutem Isolated. Należy przy nim zaznaczyć pole w kolumnie RT-CPU a wcześniej wybraną opcję odznaczyć. Aby zmiany zostały wprowadzone należy aktywować konfigurację.

CPU	RT-CPU	Base Time	CPU Limit	Latency Warning
2 (Windows)	<input type="checkbox"/>			
3 (Windows)	<input type="checkbox"/>			
4 (Windows)	<input type="checkbox"/>			
5 (Windows)	<input type="checkbox"/>			
6 (Windows)	<input type="checkbox"/>			
7 (Isolated)	<input checked="" type="checkbox"/>	1 ms	100 %	(none)

Uwaga! W niektórych systemach może nie pojawić się (po wykonaniu komendy Read from Target) rdzeń oznaczony jako (Isolated). W takiej sytuacji należy po prostu aktywować konfigurację.

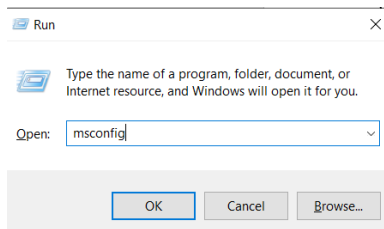
#### 2.3.4 Przywracanie wyizolowanego rdzenia

Aby przywrócić na komputerze wyizolowany rdzeń należy wywołać okno „Set on Target” (w analogiczny sposób jak przy jego izolacji). Następnie należy ustawić ilość rdzeni „Isolated” na 0 i wybrać „Set”.

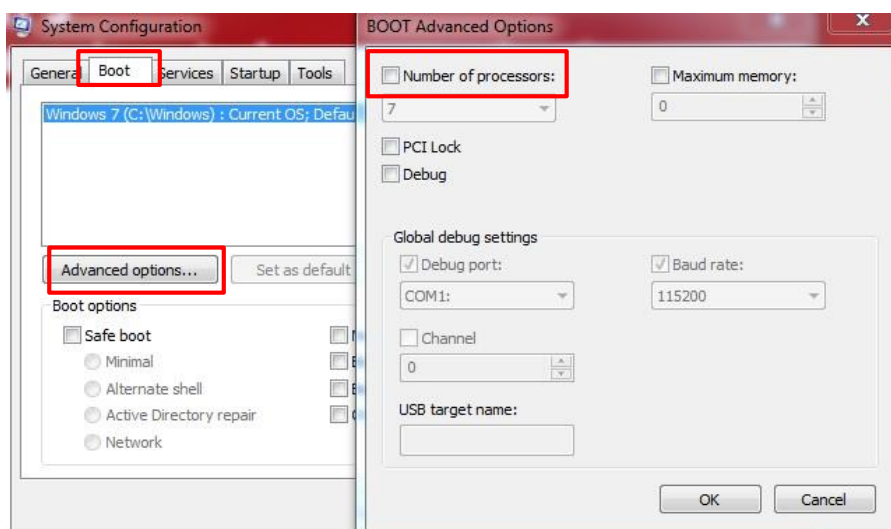


Jeśli mamy sytuację, w której rdzeń procesora nie jest widoczny w TwinCAT, przywracanie należy wykonać z poziomu systemu Windows. Wywołujemy najpierw okno Uruchom. Robimy to wpisując w Start -> Uruchom i naciskając Enter.

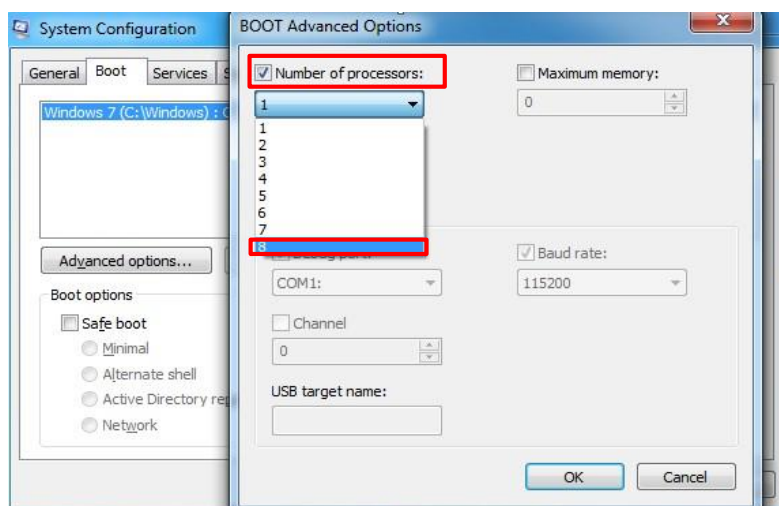
Następnie w oknie uruchom należy wpisać komendę msconfig i wybrać przycisk OK.



W oknie które się pojawi należy wybrać zakładkę **Rozruch (Boot)** i przycisk **Opcje Zaawansowane (Advanced options)**. Następnie należy odznaczyć opcję **Liczba procesorów (Number of processors)** i zatwierdzić ustawienia przyciskiem OK. W kolejnym oknie należy wybrać przycisk **Zastosuj (Apply)** a następnie **OK**. Pojawi się komunikat o konieczności restartu komputera, który należy potwierdzić.



Po restarcie komputera należy ponownie otworzyć zakładkę z ustawieniami rozruchu, zaznaczyć opcję **Number of processors** i z rozwijanej listy wybrać maksymalną ich liczbę. Zatwierdzić ustawienia w taki sam sposób jak poprzednio i ponownie zrestartować komputer. Wszystkie rdzenie będą aktywne.



### 2.3.5 Plik win8settck.bat

Jeśli po uruchomieniu obsługi wirtualizacji i wyizolowaniu rdzenia nadal pojawiają się błędy uniemożliwiające lokalne uruchomienie programu, należy uruchomić plik **win8settck.bat**. Jest to plik odpowiadający za ustawienie zegara systemowego, by umożliwić real-timeową pracę oprogramowania TwinCAT 3 na lokalnym komputerze.

Plik **win8settck.bat** znajduje się w lokalizacji:

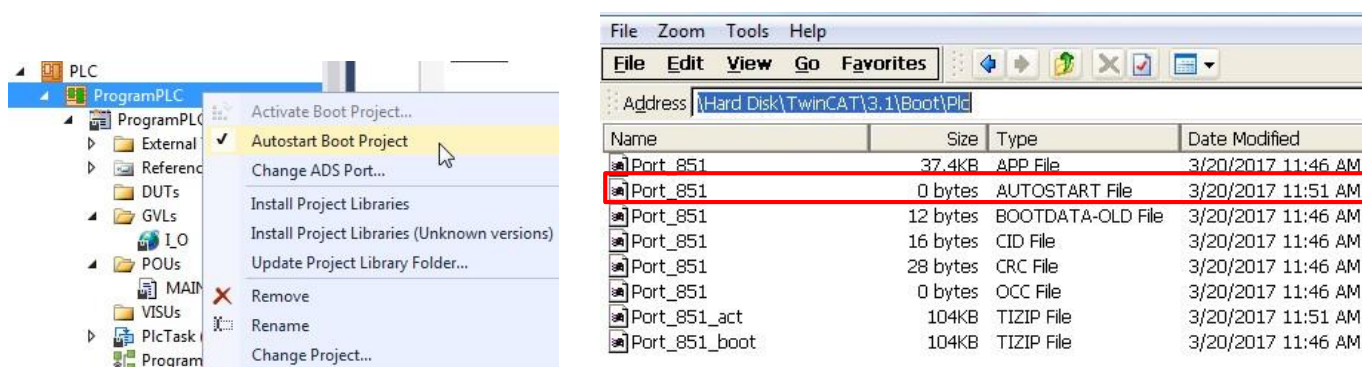
**Dysk (C: ) -> TwinCAT -> 3.1 -> System -> win8settck.bat** Należy kliknąć na ten plik PPM i **wybrać Uruchom jako administrator**.

Następnie należy zrestartować komputer.

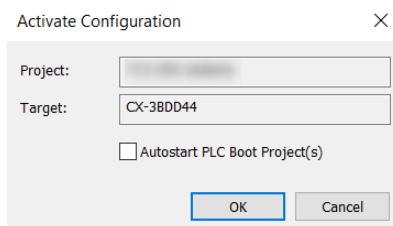
## 2.4 Projekt bootowalny, kod źródłowy na sterowniku

### 2.4.1 Wgrywanie projektu bootowalnego

Jeśli chcemy Projekt PLC ustawić jako projekt bootowalny, należy po wgraniu programu na sterownik kliknąć PPM na nazwę projektu i wybrać **Activate Boot Project** (opcja aktywna po wylogowaniu z podglądu online programu), a następnie zaznaczyć opcję **Autostart Boot Project**. Na sterowniku w lokalizacji **Hard Disk\TwinCAT\3.1\Boot\Plc** pojawi się plik który w nazwie ma numer portu i jest typu AUTOSTART FILE.



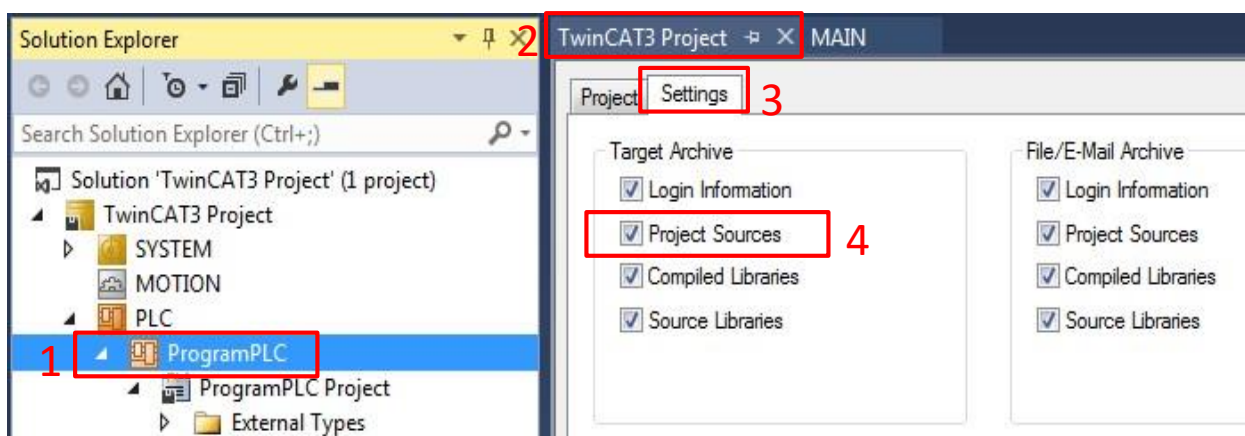
W wersji TwinCAT 3.1.4024.0 i nowszych możliwość stworzenia projektu bootowalnego pojawia się podczas procedury aktywacji konfiguracji.



### 2.4.2 Wgrywanie kodu źródłowego

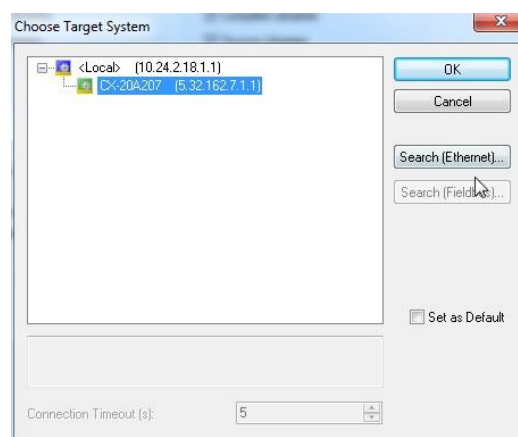
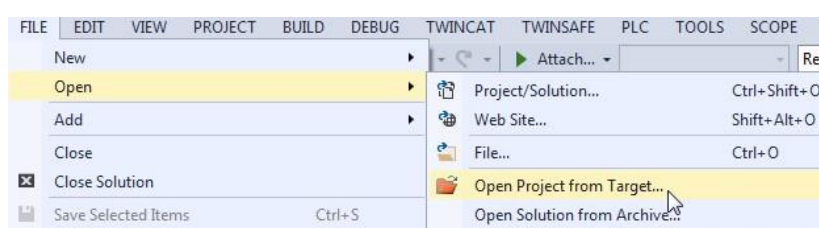
Kod źródłowy wgrywany jest automatycznie podczas wgrywania programu na urządzenie jeśli zaznaczona jest opcja **Project Sources** w ustawieniach projektu PLC. Odznaczenie tej opcji dezaktywuje automatyczne wgrywanie kodu źródłowego.

**Uwaga! Ustawienie to jest indywidualne dla każdego projektu.**



### 2.4.3 Pobieranie kodu źródłowego ze sterownika

Aby pobrać kod źródłowy oraz konfigurację z urządzenia należy wybrać **FILE -> Open -> Open Project From Target**. Następnie należy wybrać urządzenie z którego chcemy pobrać program.

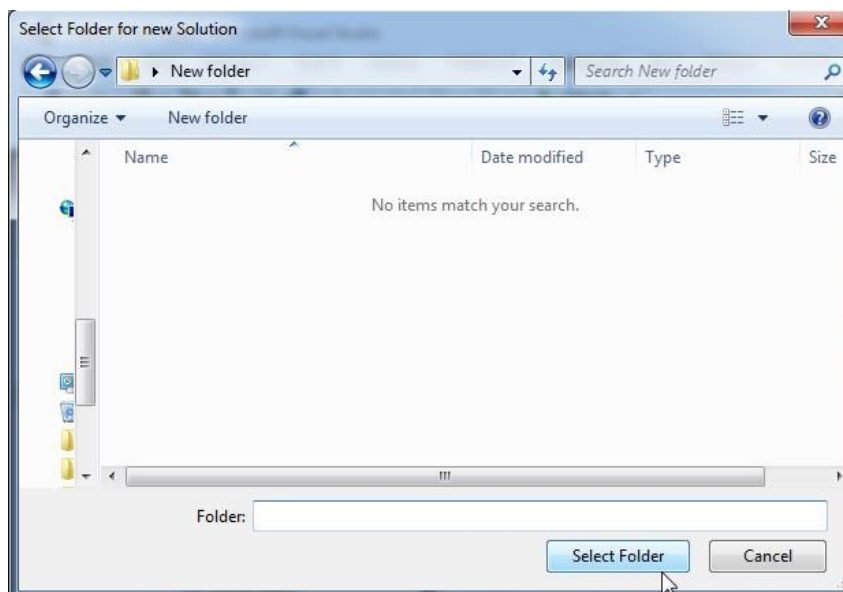


#### UWAGA!

Jeśli kod źródłowy nie został wgrany na urządzenie, to uda się pobrać jedynie konfigurację hardware oraz informację o programowych obszarach pamięci Input/Output, bez kodu programu PLC.

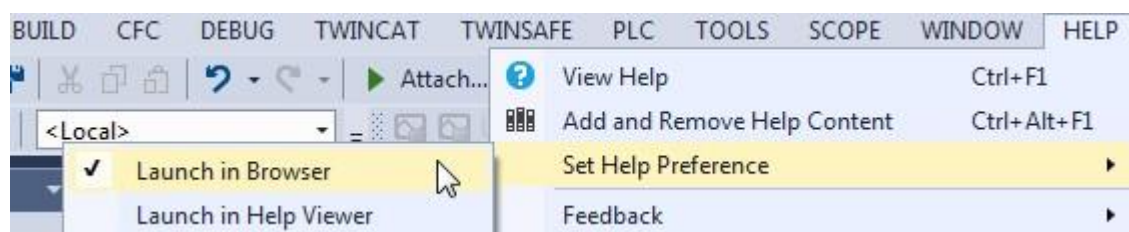
Następnie należy wskazać miejsce na dysku w którym pobrany projekt zostanie zapisany. Po tych czynnościach kod źródłowy powinien się załadować w środowisku.



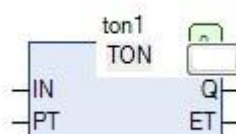


## 2.5 Offline / online Beckhoff Information System

Korzystanie z Information System bez instalacji jest możliwe jeśli komputer ma połączenie z siecią i włączone jest odpowiednie ustawienie w Visual Studio (Launch in Browser).



Przy takich ustawieniach, po zaznaczeniu nazwy bloczka i wybraniu na klawiaturze przycisku F1, opis danego bloczka otworzy nam się w wersji online Beckhoff Information System.



Aby móc lokalnie korzystać na komputerze z Beckhoff Information System bez dostępu do sieci należy pobrać plik instalacyjny ze strony [www.beckhoff.pl](http://www.beckhoff.pl).



#### Beckhoff Information System ( TwinCAT 2 included)

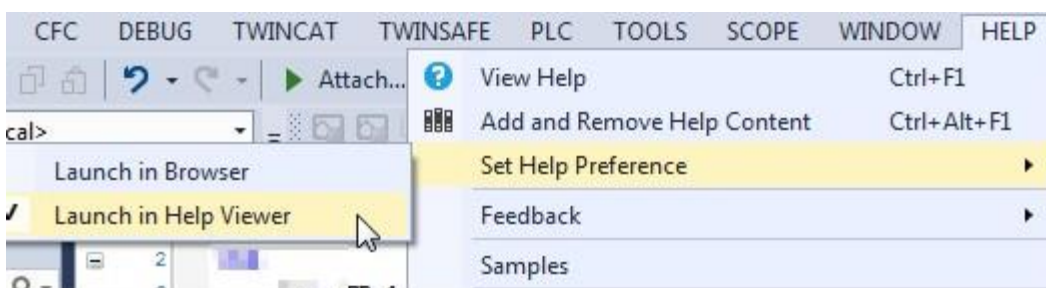
Description	Language	Version	Size	Link
Beckhoff InfoSys Installation	all	01/2017	2.0 GB	<a href="#">InfoSys.exe</a>

Furthermore the online version of the Beckhoff Information System may be found under <http://infosys.beckhoff.com>.

#### Beckhoff TwinCAT 3 Information System

Description	Language	Version	Size	Link
TwinCAT 3 Information System 32 bit and 64 bit	english	1.2.1701	253 MB	<a href="#">TC3-InfoSys.exe</a>

Po zainstalowaniu Beckhoff Information System należy zmienić ustawienie wyświetlania pomocy na Launch in Help Viewer.



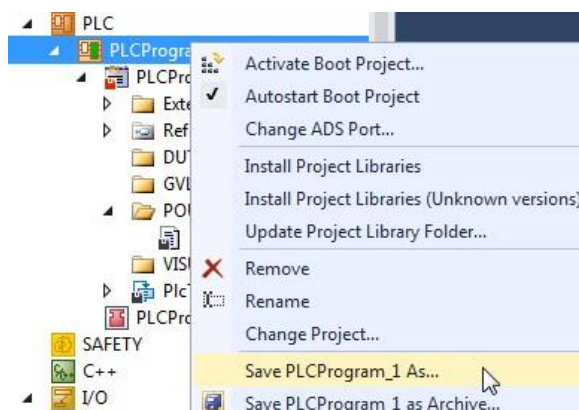
Uwaga! Zgodnie z instrukcją dostępną na stronie [infosys.beckhoff.com](http://infosys.beckhoff.com) aby zainstalować Offline Infosys w Microsoft Visual Studio należy najpierw umożliwić taką instalację. Link do instrukcji instalacji offline Infosys: [https://infosys.beckhoff.com/english.php?content=../content/1033/tc3\\_installation/72057594217401227.html](https://infosys.beckhoff.com/english.php?content=../content/1033/tc3_installation/72057594217401227.html)

### 3 Program PLC i programowanie

#### 3.1 Eksport poszczególnych części składowych projektu

##### 3.1.1 Eksport pliku .plcproj

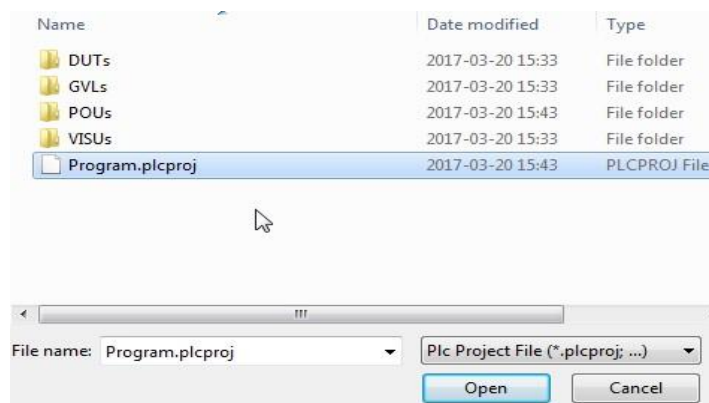
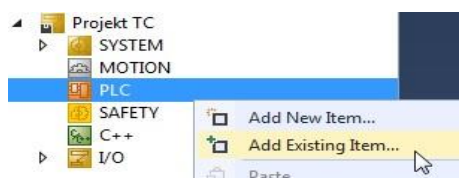
Poszczególne składowe projektu można wyeksportować i zaimportować do innych projektów. Aby wyeksportować cały Projekt PLC (zawierający zmienne globalne, typy definiowalne przez użytkownika, programy, bloki funkcyjne, funkcje etc.) należy kliknąć PPM na nazwę projektu i wybrać: **Save <nazwa projektu> As....**



Następnie należy wybrać folder w którym chcemy zapisać wyeksportowany projekt.

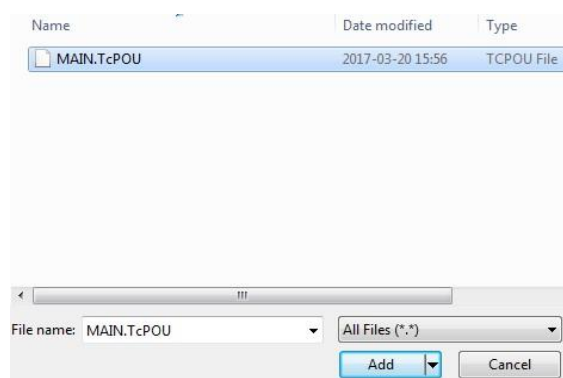
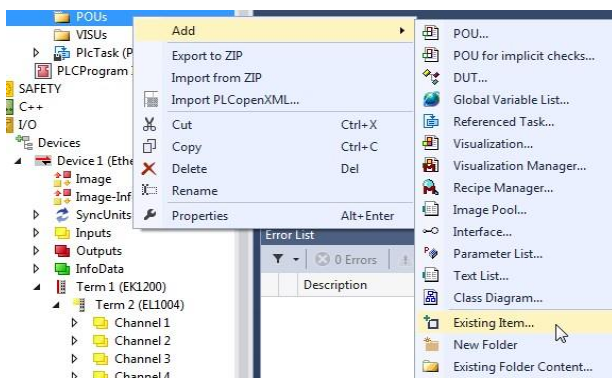
##### 3.1.2 Import poszczególnych plików składowych projektu

Aby dodać do innego Solution tak wyeksportowany projekt należy PPM kliknąć na zakładkę PLC i wybrać: **Add -> Existing Item...** Następnie z folderu do którego projekt był wyeksportowany należy wybrać plik z rozszerzeniem **.plcproj** i wybrać **Open**.



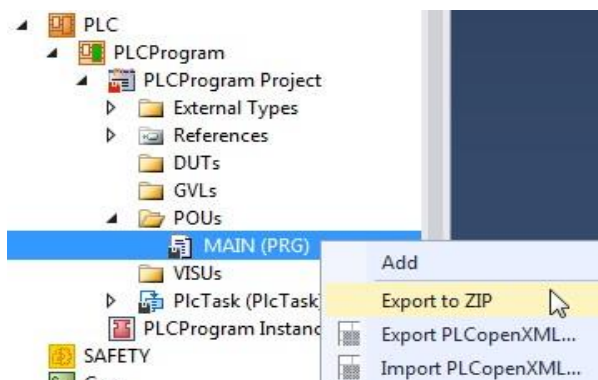
Aby dodać do projektu pojedynczy obiekt ( w przykładzie program MAIN), należy kliknąć w odpowiednim miejscu w drzewie projektu PPM i dodać **Add -> Existing Item...** Ponieważ program wyeksportowany w przykładzie powyżej zapisuje się wraz z folderem do którego przynależy, trzeba odnaleźć plik z rozszerzeniem **.TcPOU**





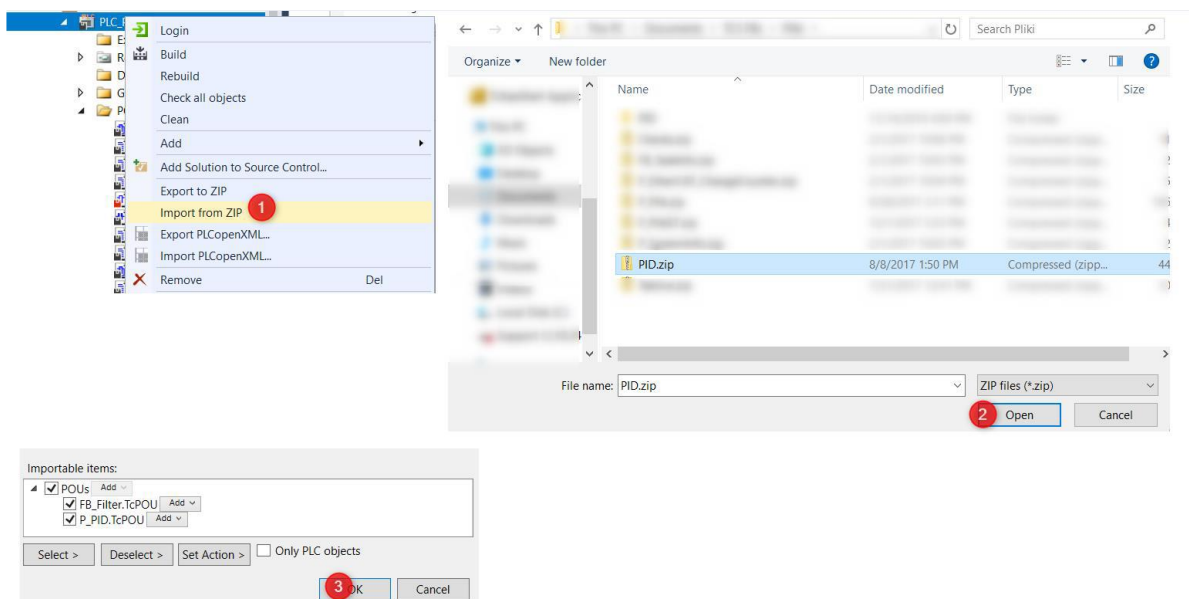
### 3.1.3 Eksport do pliku .zip

Można również eksportować pojedyncze instancje programów, bloków funkcyjnych etc. Należy kliknąć PPM na wybranym obiekcie a następnie wybrać **Export to ZIP** i wybrać lokalizację gdzie pliki mają zostać wyeksportowane.



### 3.1.4 Import z pliku .zip

Aby zaimportować pliki projektu wyeksportowane do pliku .zip należy kliknąć PPM w odpowiednim miejscu w drzewku projektu i wybrać opcję **Import from ZIP** a następnie wybrać odpowiedni plik i w kolejnym oknie potwierdzić wybór żądanych plików, które mają zostać zaimportowane.

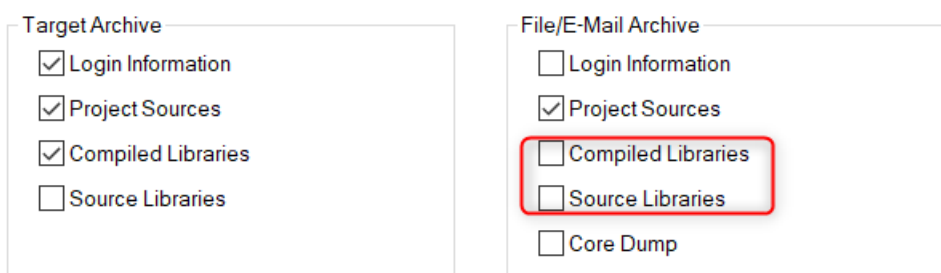


### 3.1.5 Eksport w formacie .tnzip

TwinCAT 3 pozwala na eksport całego rozwiązania (czyli zarówno TwinCAT Project, jak i np. Measurement Project i DataBase Project, o ile znajdują się w jednym rozwiązaniu) jako plik .tnzip. Plik ten następnie można bez problemu otworzyć i otrzymamy pełne rozwiązanie zawierające wszystkie składowe. Aby wyeksportować projekt w formacie .tnzip należy wybrać z górnego menu **File -> Save <nazwa solution> as Archive** i wybrać miejsce, w którym chcemy zapisać projekt.

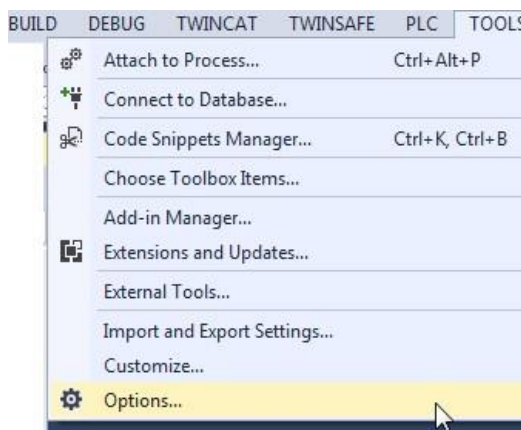
Aby otworzyć projekt zapisany w formacie .tnzip należy wybrać **File -> Open -> Open Solution from Archive** i wybrać odpowiedni plik .tnzip, a następnie wybrać folder, do którego rozpakowane zostanie całe Solution.

Aby zmniejszyć rozmiar plik .tnzip, należy w ustawieniach projektu, w sekcji **File/E-Mail Archive** odznaczyć „Compiled Libraries” i „Source libraries” zostawiając tylko „Project Sources”.

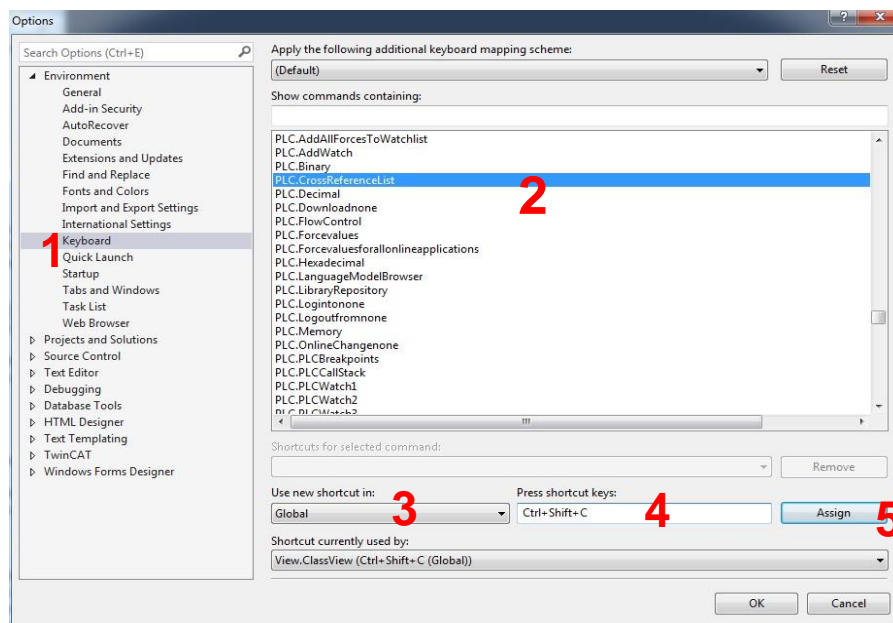


## 3.2 Definiowanie własnych skrótów klawiszowych

Aby przyspieszyć pracę z TwinCAT 3 można zdefiniować własne skróty klawiszowe. Aby to zrobić należy z górnego menu wybrać **TOOLS -> Options**



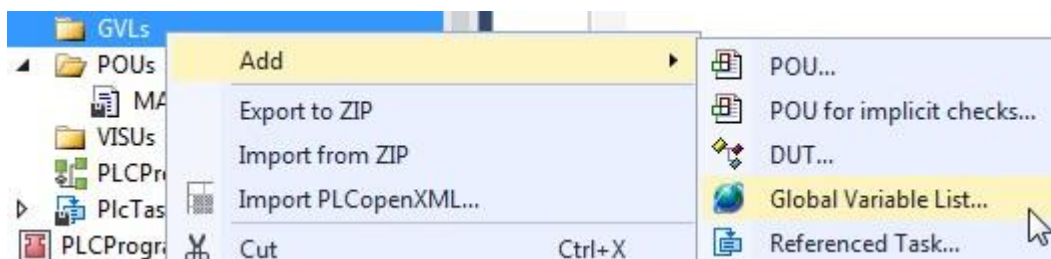
Następnie w oknie **Options** po lewej stronie należy wybrać pozycję **Keyboard(1)**. Z listy komend(2) należy wybrać tę do której chcemy utworzyć skrót klawiszowy. W polu **Use new shortcut in(3)** należy wybrać część środowiska w której skrót będzie obowiązywać (ustawienie Global odnosi się do całego środowiska). Następnie należy ustawić kursor w polu **Press shortcut keys(4)** i wcisnąć na klawiaturze wybrany przez siebie skrót klawiszowy. Wybór należy zatwierdzić przyciskiem **Assign(5)**.



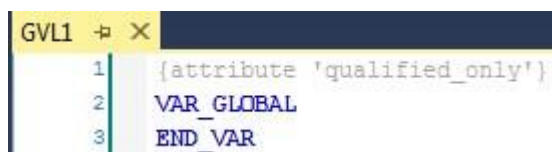
### 3.3 Dostęp do zmiennych

#### 3.3.1 Zmienne globalne

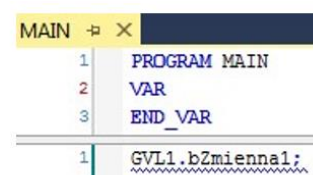
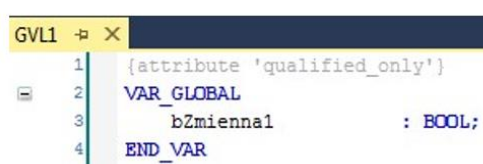
Aby dodać listę zmiennych globalnych należy w drzewie projektu PPM kliknąć na folder GVLs i wybrać **Add** -> **Global Variable List** a następnie nadać nazwę listy.



Pusta lista zmiennych globalnych wygląda następująco:



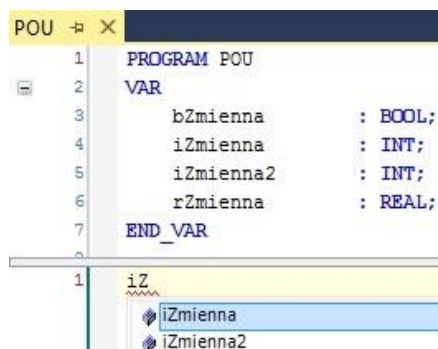
Lista zmiennych globalnych posiada atrybut **'qualified\_only'**. Po zadeklarowaniu zmiennej globalnej, odwołanie do niej w programie odbywa się poprzez podanie nazwy listy zmiennych globalnych i po kropce odwołanie do odpowiedniej zmiennej.



### 3.3.2 Zmienne lokalne

Zmienne lokalne deklaruje się pomiędzy znacznikami **VAR** oraz **END\_VAR**. Odwołanie do takiej zmiennej odbywa się bezpośrednio poprzez jej nazwę, ale tylko w programie w którym została zadeklarowana.

Podczas używania zmiennej w programie można skorzystać z podpowiedzi wywoływanych za pomocą skrótu klawiszowego Ctrl+Spacja.



Podczas deklaracji zmiennych można je podzielić na sekcje, w celu uporządkowania danych. Dane sekcje można ukrywać/pokazywać.

```

2  VAR GLOBAL
3  (*wejścia cyfrowe*)
4      bPrzycisk1      AT%I+      :BOOL;
5      bPrzycisk2      AT%I+      :BOOL;
6      bPrzycisk3      AT%I+      :BOOL;
7      bPrzycisk4      AT%I+      :BOOL;
8
9  (*wyjścia cyfrowe*)
10     bLampka1         AT%Q+      :BOOL;
11     bLampka2         AT%Q+      :BOOL;
12     bLampka3         AT%Q+      :BOOL;
13     bLampka4         AT%Q+      :BOOL;
14
15  (*wejścia analogowe*)
16     iPotencjometr1    AT%I+      :INT;
17     iPotencjometr2    AT%I+      :INT;
18
19  (*wyjścia analogowe*)
20     iWyjscieAnalogowe1 AT%Q+      :INT;
21     iWyjscieAnalogowe2 AT%Q+      :INT;
22  END_VAR

```

### 3.4 Wywoływanie bloków funkcyjnych

#### 3.4.1 Wywołanie bloku funkcyjnego w języku ST

Aby wywołać blok funkcyjny w języku tekstowym najlepiej w polu kodu PLC wywołać Input Assistant poprzez wciśnięcie F2, następnie wyszukać interesujący nas blok i wybrać OK **pamiętając o zaznaczeniu checkboxa „Insert with arguments”**. Następnie należy zadeklarować blok funkcyjny i wtedy pojawi się on ze wszystkimi wejściami i wyjściami, które wystarczy przypisać (lub usunąć jeśli chcemy pozostawić je na wartościach domyślnych).

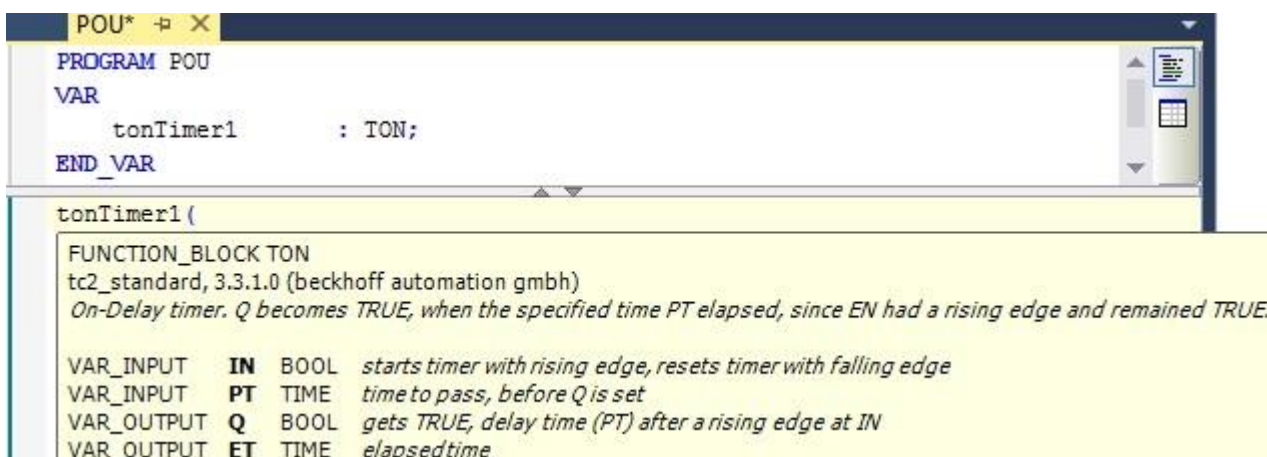
Structured view

Documentation:

☒ Insert with arguments

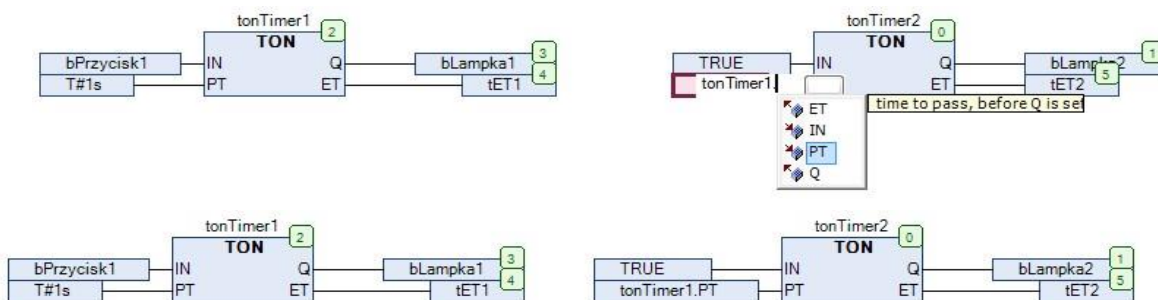
```
TimerTON1(IN:= , PT:= , Q=> , ET=> );|
```

Można też w polu deklaracji zadeklarować jego nazwę i typ. Wtedy po użyciu nazwy bloku w polu kodu programu, pojawi się podpowiedź o dostępnych wejściach i wyjściach bloku.



### 3.4.2 Przekazywanie wartości z bloku funkcyjnego

Jeżeli chcemy przekazywać wartości z wejść/wyjść konkretnego bloku funkcyjnego w inne miejsce programu, można się do odwoływać do wejść/wyjść bloku poprzez podanie nazwy bloku i po kropce nazwy konkretnego wejścia/wyjścia.



Bloczek Timera tonTimer2 będzie na swoje wejście PT pobierał wartość podaną na wejście PT bloczka Timera tonTimer1. Po zadeklarowaniu bloku funkcyjnego w programie można w trybie online rozwinąć jego pełną strukturę.

TwinCAT_Device.ProgrsmPLC.POU			
Expression	Type	Value	Comment
tonTimer	TON		
IN	BOOL	FALSE	starts timer with rising...ge, resets timer with...
PT	TIME	T#0ms	time to pass, before Q is set
Q	BOOL	FALSE	gets TRUE, delay time...) after a rising edge...
ET	TIME	T#0ms	elapsed time
M	BOOL	FALSE	
StartTime	TIME	T#0ms	

Odwołując się w kodzie programu do danego elementu struktury bloku, jako podpowiedzi wyświetlają się jedynie pola oznaczone jako wejścia i wyjścia, nie są widoczne zmienne wewnętrzne (w przypadku timera TON: M oraz StartTime), można ich jednak również używać.



```

1 PROGRAM POU
2 VAR
3     tonTimer          : TON;
4 END_VAR
5
6 tTime:=tonTimer.

```

time to pass, before Q is set

```

MAIN POU
1 PROGRAM POU
2 VAR
3     tonTimer          : TON;
4     tTime             : TIME;
5     tStartTime        : TIME;
6 END_VAR
7
8 tTime:=tonTimer.PT;
9
10 tStartTime:=tonTimer.StartTime;

```

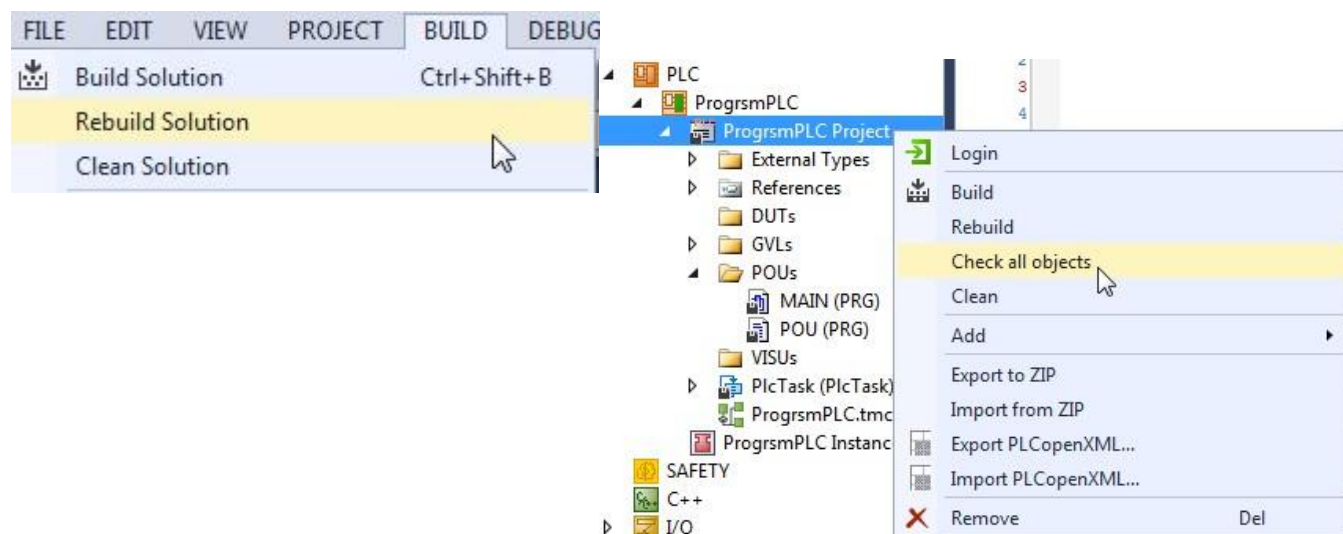
## 3.5 Debugowanie

### 3.5.1 Sprawdzanie wszystkich składowych projektu

Jeśli w projekcie znajduje się kilka programów, bloków funkcyjnych etc. to podczas przybudowania projektu (BUILD -> Rebuild Solution) sprawdzane są tylko te składowe projektu, które są wywołane. Przykład:

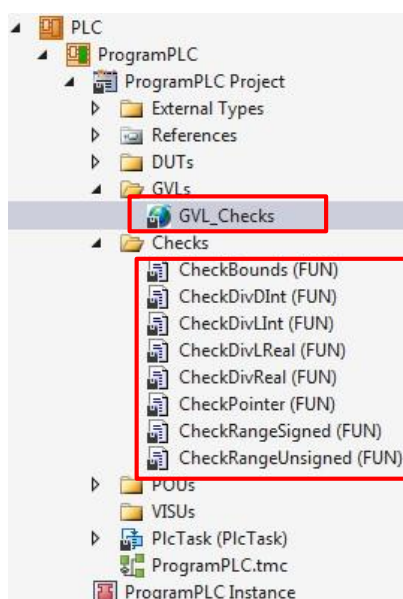
W projekcie znajdują się dwa programy: MAIN oraz POU. Program POU nie jest używany (nie jest wywołany) ale znajdują się w nim błędy składni. Po przebudowaniu projektu nie pojawiają się komunikaty o błędach. Jeśli chcemy sprawdzać również składnię w niewywołanych elementach, należy PPM myszy kliknąć na nazwie projektu i wybrać opcję **Check All Objects**.

Pojawią się wtedy komunikaty o błędach, które występują w niewywołanych elementach projektu.



### 3.5.2 Checks

Do diagnostyki aplikacji, na etapie testowania, można wykorzystywać funkcje systemowe – **Checks**. Są to funkcje sprawdzające czy w projekcie nie występują próby dzielenia przez zero oraz czy nie przekraczamy indeksów tablic. Aby móc z nich korzystać, należy je dodać do swojego projektu, ale nie należy ich już wywoływać (PPM -> Add Existing Item). Po uruchomieniu programu, należy otworzyć listę zmiennych globalnych **GVL\_Checks** i sprawdzić czy któryś z liczników nie inkrementuje się.



TwinCAT_Device.ProgramPLC.GVL_Checks			
Expression	Type	Value	Comment
dCheckBounds	DINT	0	Licznik przekroczen zakresu indexow, np. tablic
dCheckDivDInt	DINT	2819	Licznik prob dzielenia Dint przez 0
dCheckDivLInt	DINT	0	Licznik prob dzielenia LInt przez 0
dCheckDivLReal	DINT	0	Licznik prob dzielenia LREAL przez 0
dCheckDivReal	DINT	0	Licznik prob dzielenia REAL przez 0
dCheckRangeS...	DINT	0	Licznik przekroczen liczby znakowej
dCheckRangeU...	DINT	0	Licznik przekroczen liczby bezznakowej

Jak widać na powyższym przykładzie, w diagnozowanym projekcie inkrementuje się licznik prób dzielenia przez zero zmiennej typ Integer.

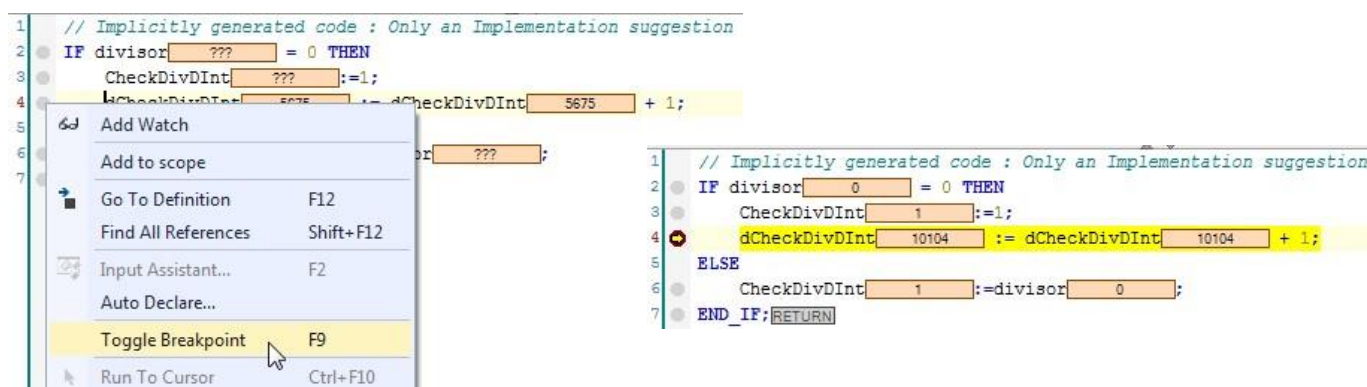
W celu odnalezienia miejsca w programie, gdzie występują próby dzielenia przez zero, należy odnaleźć funkcję odpowiadającą za inkrementowanie się danego licznika. W przykładzie inkrementuje się licznik **dCheckDivDInt**, zatem odpowiadającą mu funkcją jest funkcja **CheckDivDInt**.

### 3.5.3 Breakpoints

#### 3.5.3.1 Określanie breakpointów

Breakpointy są miejscami określonymi w kodzie, w momencie wykonania których program PLC natychmiast się zatrzymuje. Pozwalają one np. na zabezpieczenie przed wejściem w niepożądaną część programu czy, jak w przytoczonym przykładzie, pomagają w obłudze innych funkcji. Działanie breakpointów opisane zostanie na podstawie wcześniej opisanych funkcji Checks.

Po określeniu, która funkcja Checks odpowiada za naliczanie wartości licznika w GVL, należy otworzyć tę funkcję i w linii kodu odpowiadającej za inkrementowanie licznika określić breakpoint. Aby określić breakpoint należy w trybie online postawić kursor myszy w wybranej linii, PPM nacisnąć na wybraną linię i wybrać opcję Toggle Breakpoint.



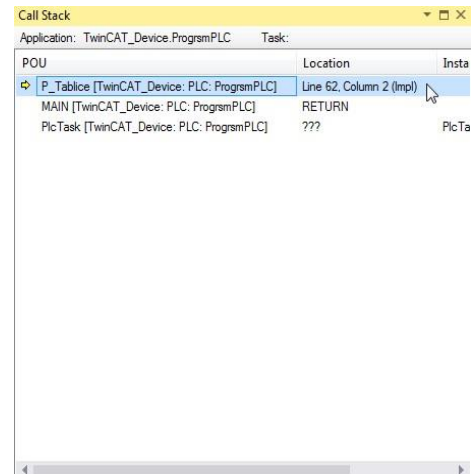
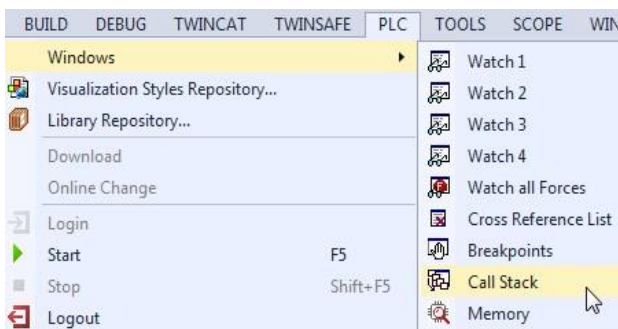
**BREAKPOINT ZATRZYMUJE PROGRAM PLC!**



### 3.5.3.2 Wywołanie stosu

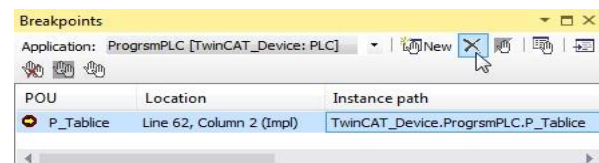
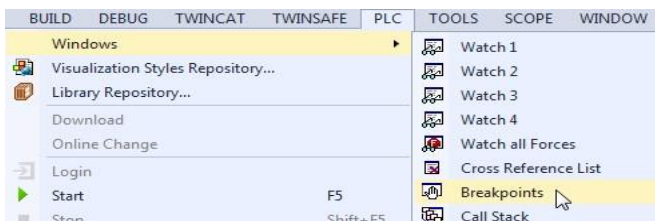
Aby sprawdzić do których elementów projektu odwołuje się linia kodu gdzie został określony breakpoint, należy wywołać okno **Call Stack**.

Po wybraniu odpowiedniej pozycji z listy **Call Stack** i dwukrotnym kliknięciu LPM zostaniemy przeniesieni do odpowiedniego miejsca w projekcie. Należy poprawić kod i ponownie sprawdzić listę zmiennych globalnych zawierających liczniki checks.

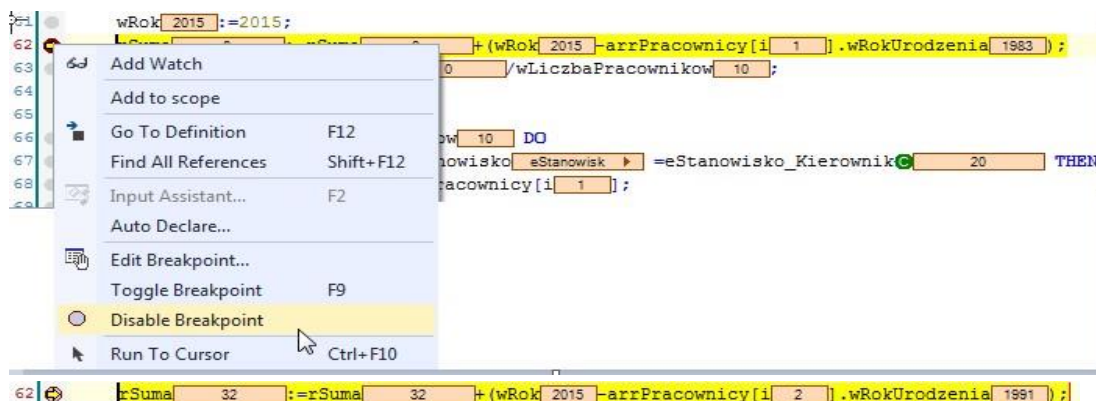


### 3.5.3.3 Zarządzanie breakpointami

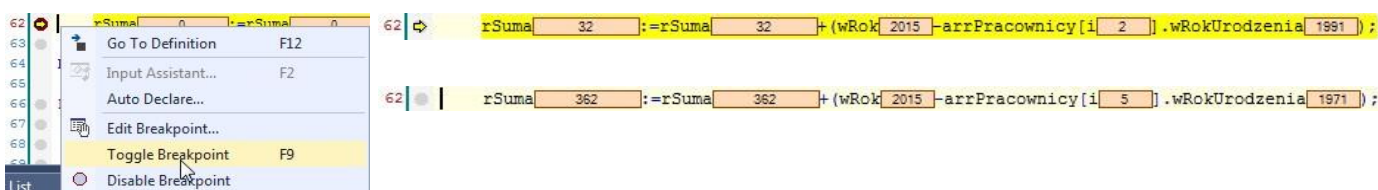
Aby wywołać okno z którego można zarządzać wszystkimi breakpointami w projekcie, należy wybrać **PLC -> Windows -> Breakpoints**. Po wybraniu Breakpointa z listy, w oknie Breakpoints, można go z tego poziomu usunąć.



Aby wyłączyć aktywny Breakpoint należy kliknąć PPM i wybrać opcję Disable Breakpoint i uruchomić program.

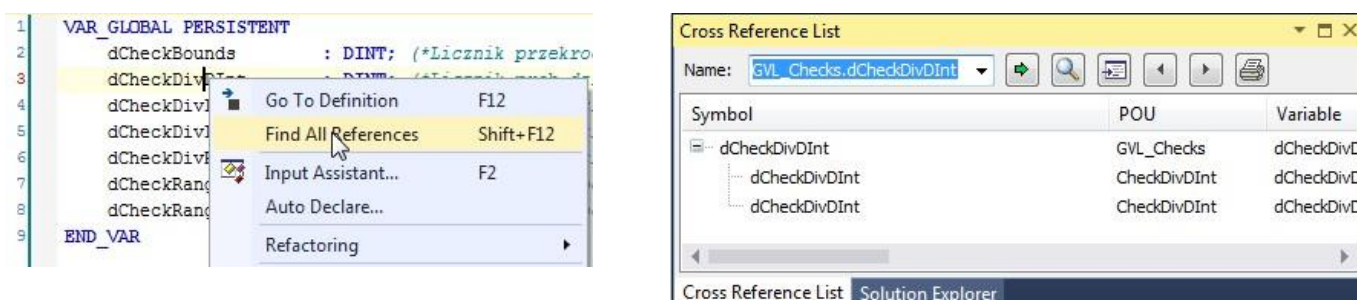


Aby usunąć aktywny breakpoint należy kliknąć PPM myszy na linię w której się on znajduje i ponownie wybrać opcję Toggle Breakpoint i uruchomić program.



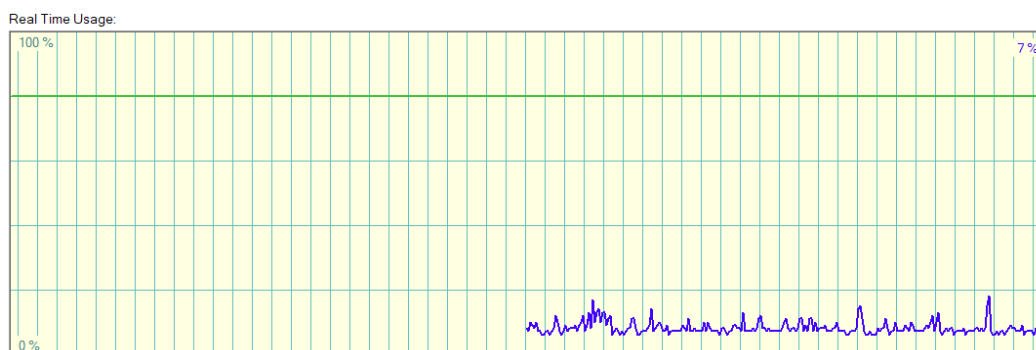
### 3.5.4 Znajdowanie referencji w projekcie

Aby odnaleźć wszystkie odwołania do danej zmiennej w programie (listę referencji) należy postawić kursor na nazwie zmiennej i kliknąć PPM a następnie wybrać opcję **Find All References**. Po dwukrotnym kliknięciu LPM zostaniemy przeniesieni do odpowiedniego miejsca w projekcie.



### 3.5.5 Sprawdzanie obciążenia sterownika

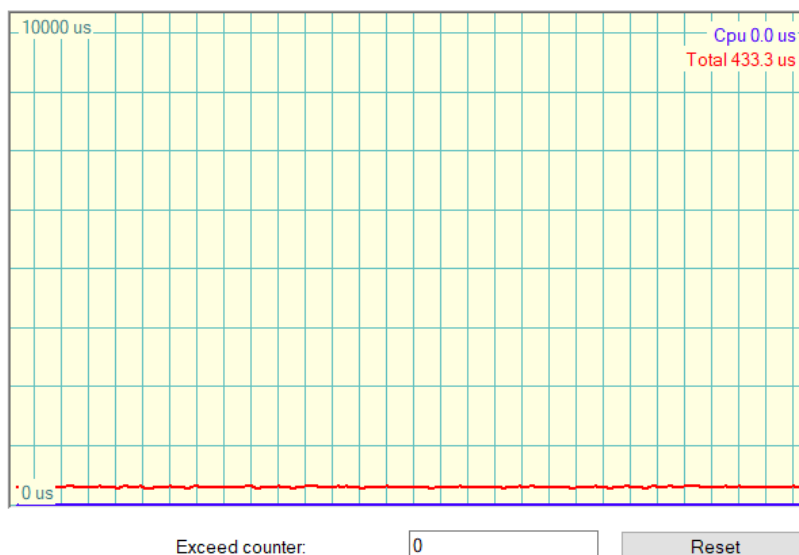
Aby sprawdzić aktualne obciążenie Run-Time sterownika, należy w drzewie projektu przejść do **System -> Real-Time**, następnie wybrać tam zakładkę **Online**. Obciążenie sterownika będzie tam przedstawione w postaci wykresu rysowanego w czasie rzeczywistym.



### 3.5.6 Sprawdzanie czasu wykonywania tasku

#### 3.5.6.1 Sprawdzanie z poziomu drzewa projektu

Aby sprawdzić aktualny czas wykonywania tasku, należy w drzewie projektu przejść do **System -> Tasks -> Nasz wybrany Task**, a następnie wybrać tam zakładkę **Online**. Aktualny czas wykonywania tasku w stosunku do maksymalnego możliwego czasu przedstawiony będzie w postaci wykresu rysowanego w czasie rzeczywistym. Dostępny będzie tam też licznik **Exceed Counter**, który zlicza ilość przekroczeń czasu wykonywania tasku. W poprawnie działającym projekcie licznik ten powinien wskazywać wartość 0. Każde przekroczenie czasu wykonywania skutkuje utraceniem działania programu w czasie rzeczywistym.



### 3.5.6.2 Sprawdzanie z poziomu programu PLC

Czas wykonywania tasku (oraz dodatkowe informacje) można sprawdzić także z poziomu programu PLC. Służy do tego specjalny blok funkcyjny FB\_TaskInfo (dostępny po kontakcie z działem technicznym pod adresem [support@beckhoff.pl](mailto:support@beckhoff.pl)). Blok ten należy wywołać **w każdym tasku który chcemy monitorować**. Po wywołaniu bloku w zakładce online można sprawdzić informacje o tasku takie jak np.

- nazwa tasku i priorytet
- najdłuższy czas wykonania
- średni czas wykonania
- łączny czas pracy
- czas wykonania określonego cyklu, i inne...

fbTaskInfo	FB_TaskInfo				
bReset	BOOL	FALSE			
bError	BOOL	FALSE			
fbGetCurTaskIdx	GETCURTASKIN...				
rLastExecTime	REAL	0.0995			
rCycleTime	REAL	10			
uiPriority	UINT	20			
sTaskName	STRING(63)	'PLC_Project_Pl...			
bCycleTimeExceeded	BOOL	FALSE			
rCycleCount	REAL	1341757			
rTotalTime	REAL	184286.359			
rAverageCycleTime	REAL	0.137347043			
bResetVisu	BOOL	FALSE			

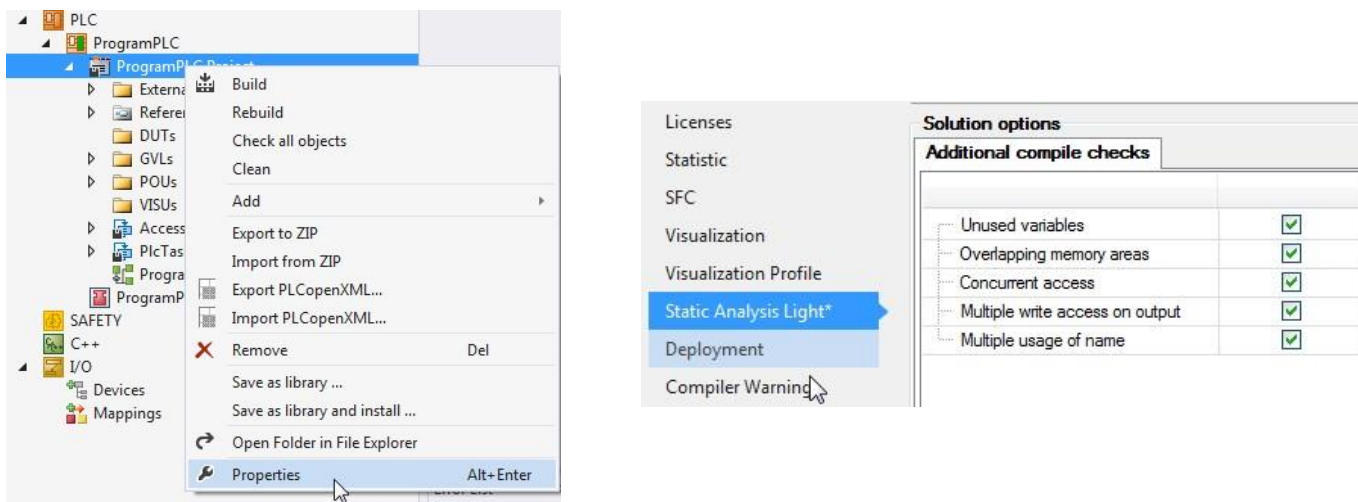
### 3.5.7 Dodatkowe opcje debuggowania

Przy domyślnych ustawieniach środowiska TwinCAT 3 nie są sprawdzane następujące „błędy”:

- **Unused variables** – nieużywane zmienne
- **Overlapping memory areas** – nakładające się obszary pamięci
- **Concurrent access** – dostęp do zmiennej z różnych tasków
- **Multiple write acces on output** – nadpisywanie wyjścia z różnych miejsc

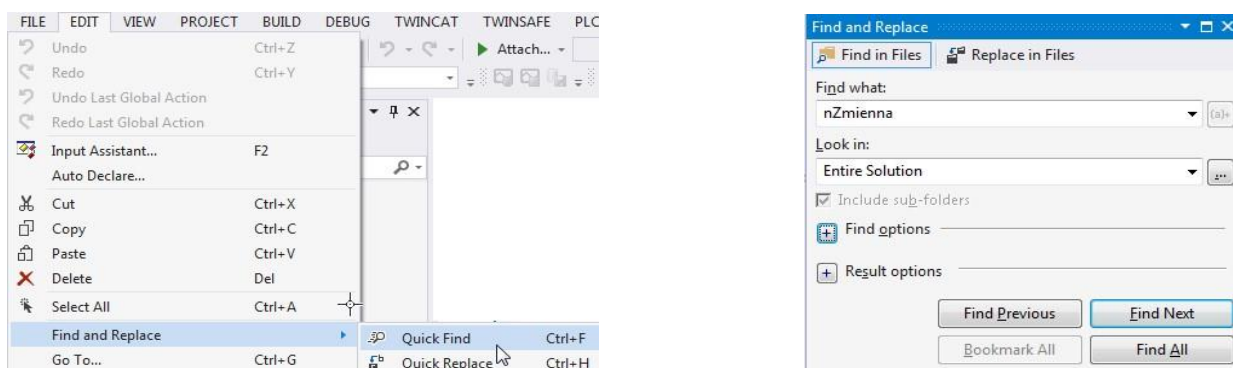
- **Multiple usage of name** – użycie tej samej nazwy w różnych miejscach

Jeśli chcemy aktywować wyżej wymienione opcje należy uaktywnić okno właściwości projektu. W tym celu należy kliknąć PPM na nazwie projektu i wybrać **Properties**, a następnie przejść do zakładki **Static Analysis Light** i zaznaczyć interesujące nas opcje.

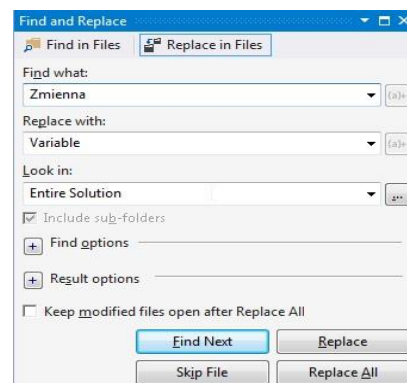
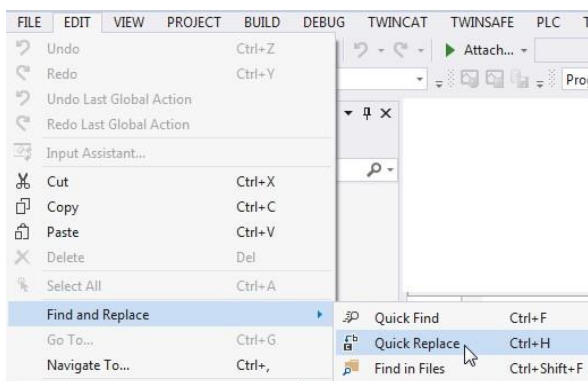


### 3.6 Find & replace

Jeśli chcemy wyszukać daną frazę w projekcie należy wybrać **EDIT -> Find and Replace -> Quick Find**. W oknie dialogowy, które się pojawi w polu „**Find what**” wpisujemy szukaną frazę, a w polu „**Look in**” wybieramy jaki obszar Solution chcemy przeszukiwać.



Wyszukując daną frazę w Solution, możemy ją zastąpić inną frazą. Należy wtedy wybrać **EDIT -> Find and Replace -> Quick Replace**. W oknie dialogowym w polu „**Find what**” wpisujemy szukaną frazę, w polu „**Replace with**” wpisujemy frazę którą chcemy zastąpić wyszukany ciąg liter a w polu „**Look in**” określamy obszar poszukiwań.



## 3.7 Komentarze

### 3.7.1 Komentarze w języku ST

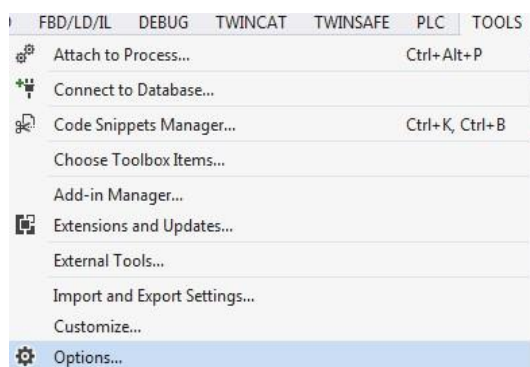
Komentarze w polu deklaracji zmiennych dla wszystkich języków oraz w polu kodu języka ST można umieszczać po znakach **//komentarz** (komentarz dotyczy aktualnej linii) lub pomiędzy znakami **(\* komentarz \*)** (komentarz wielolinijkowy).

```

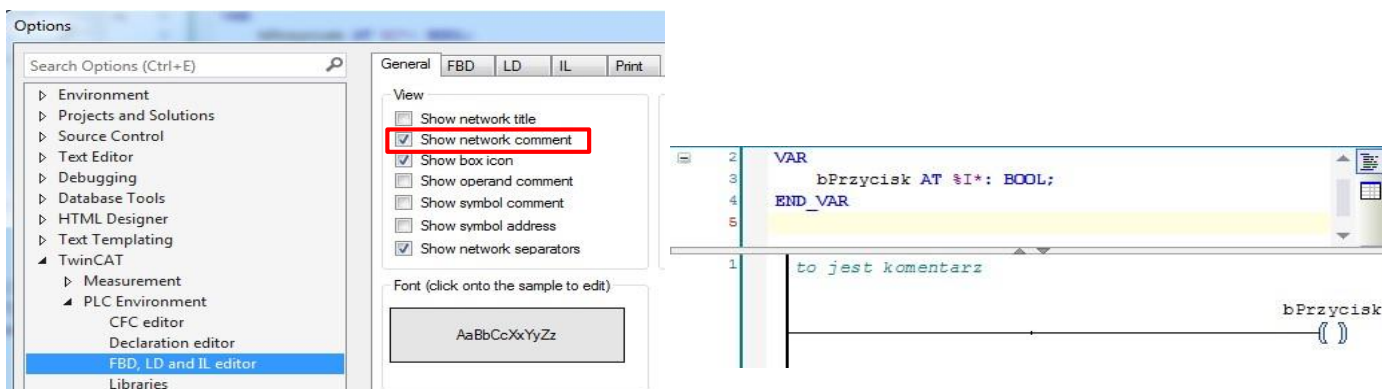
1  PROGRAM POU_1
2  VAR
3
4      bZmienna      :BOOL;  (*zmienna boolowska
5                               rozmiar 1 bajt *)
6  END_VAR
7
8  //iZmienna;
```

### 3.7.2 Komentarze w językach IL, LD, FBD

Aby móc umieszczać komentarze w poszczególnych liniach kodu w językach IL, LD oraz FBD należy najpierw uaktywnić taką opcję w ustawieniach TwinCAT. Najpierw należy wybrać **TOOLS -> Options**.



Następnie odszukujemy opcje dotyczące **TwinCAT -> PLC Enviroment -> FBD, LD and IL editor**. Znajduje się tam ustawienie **Show network comment**, które należy zaznaczyć. Od tej pory możemy wstawiać komentarze w językach IL, LD i FBD w górnej części linii kodu. Wystarczy ustawić tam kursor i wpisać komentarz.

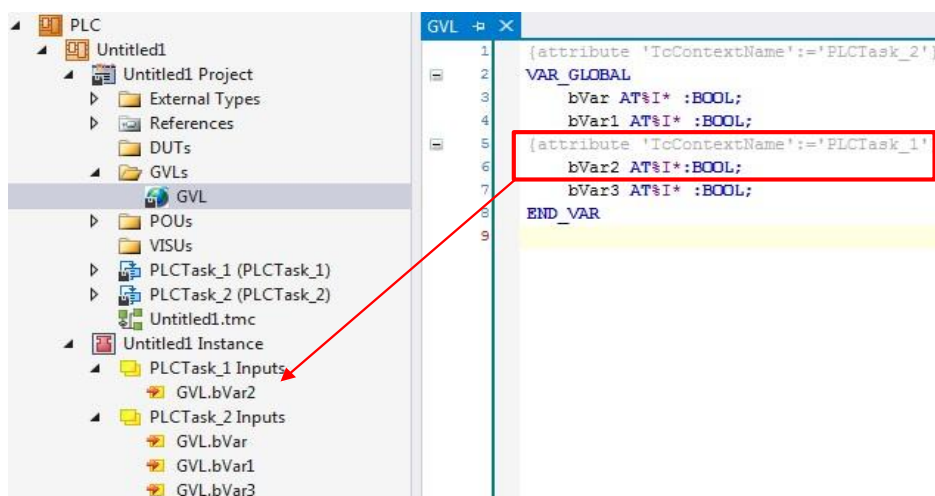


## 3.8 Atrybuty

### 3.8.1 TcContextName

Atrybut `TcContextName` pozwala na przydzielenie konkretnych zmiennych do poszczególnych tasków w programach które posiadają więcej niż jeden task. W przykładzie mamy skonfigurowane dwa taski: `PLCTask_1` oraz `PLCTask_2`. Tworzymy jedną listę zmiennych globalnych dla całego projektu i przypisujemy zmiennym z tej listy atrybuty, które przyporządkowują je do odpowiednich tasków.

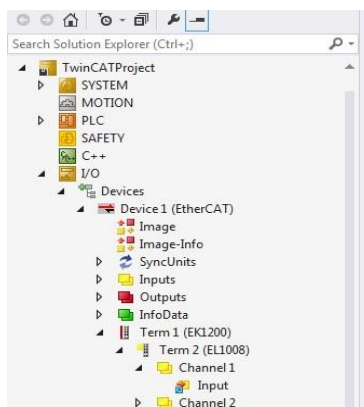
W przykładzie atrybuty są tak skonfigurowane, że wszystkie zmienne, poza zmienną `bVar2`, będą przypisane do `PLCTask_2`, ponieważ atrybut przypisujący do `PLCTask_2` znajduje się przed całą listą.



### 3.8.2 TcLinkTo

Atrybut `TcLinkTo` pozwala na linkowanie zmiennych z poziomu pola deklaracji zmiennych, bez potrzeby robienia tego z poziomu konfiguracji hardware. W podanym przykładzie zmienna została zlinkowana do kanału 1 modułu `EL1008`.





```

1
2
3 VAR GLOBAL
4   {attribute 'ToLinkTo':='TIID' Device 1 (EtherCAT)^Term 1 (EK1200)^Term 2 (EL1008)^Channel 1^Input'}
5   bPrzycisk1    AT%I* :BOOL;
6   bLampka1      AT%Q* :BOOL;
7   iPotencjometr1 AT%I* :INT;
8   iWyjscieAnalogowe AT%Q* :INT;
9 END_VAR

```

### 3.8.3 Displaymode

Każdej zmiennej można przypisać atrybut określający w jakiej reprezentacji ma być przedstawiana jej wartość – binarnej, decymalnej lub hexadecymalnej.

W przedstawionym przykładzie 3 zmienne zostały nałożone na ten sam obszar pamięci ale każda ma przypisany inny atrybut. W efekcie mamy jedną wartość wyświetloną w 3 różnych formatach:

```

PROGRAM MAIN
VAR
  {attribute 'displaymode':='bin'}
  iZmienna1 AT%MB100:INT;
  {attribute 'displaymode':='dec'}
  iZmienna2 AT%MB100:INT;
  {attribute 'displaymode':='hex'}
  iZmienna3 AT%MB100:INT;
END_VAR

```

Expression	Type	Value	Prepa...	Address
iZmienna1	INT	2#0000000011010111		%MB100
iZmienna2	INT	215		%MB100
iZmienna3	INT	16#00D7		%MB100

### 3.8.4 Hide

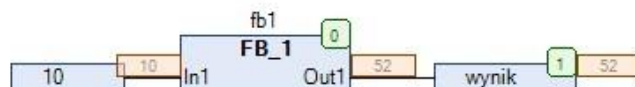
Atrybut 'Hide' pozwala określić które zmienne mają być niewidoczne, np. niektóre zmienne lokalne w blokach funkcyjnych. W przykładzie, wewnątrz bloku są zadeklarowane dwie zmienne lokalne, jedna z nich (Local2) ma przypisany atrybut 'Hide'. Po wywołaniu bloku w programie i zalogowaniu, zmienna Local 2 nie będzie widoczna w strukturze bloku.

```

1 FUNCTION_BLOCK FB_1
2   VAR_INPUT
3     In1 : INT;
4   END_VAR
5   VAR_OUTPUT
6     Out1 : INT;
7   END_VAR
8   VAR
9     Local1 : INT:=5;
10    {attribute 'hide'}
11    Local2 : INT:=2;
12  END_VAR
13
14  Out1:=In1*Local1+Local2;

```

MAIN [Online] FB_1 [Online]				
TwinCAT_Device.Untitled1.MAIN				
Expression	Type	Value	Prepa...	Address
fb1	FB_1			
In1	INT	10		
Out1	INT	52		
Local1	INT	5		
wynik	INT	52		



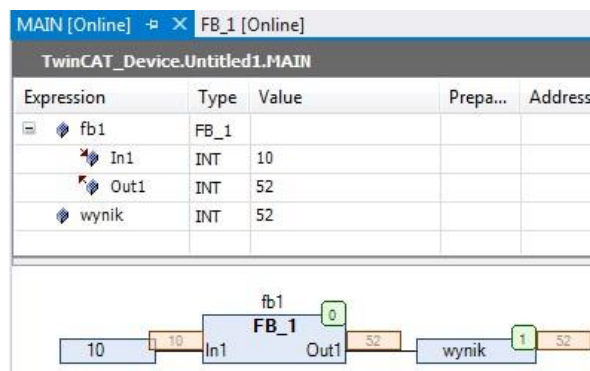
### 3.8.5 Hide\_all\_locals

Atrybut 'Hide all Locals' pozwala ukryć wszystkie zmienne lokalne z bloku, tak aby po wywołaniu programu w bloku zmienne nie były widoczne w strukturze bloku.

```

1 {attribute 'hide_all_locals'}
2 FUNCTION_BLOCK FB_1
3 VAR_INPUT
4     In1 : INT;
5 END_VAR
6 VAR_OUTPUT
7     Out1 : INT;
8 END_VAR
9 VAR
10     Local1 : INT:=5;
11     Local2 : INT:=2;
12 END_VAR
13
14 Out1:=In1*Local1+Local2;

```



### 3.8.6 Strict

Atrybut **Strict** dodaje się automatycznie podczas tworzenia typu wyliczeniowego (ENUM). Atrybut ten nie dopuszcza do wykonywania operacji arytmetycznych na zmiennej ENUM. Chroni to np. przed sytuacją, kiedy w wyniku działań arytmetycznych na zmiennej ENUM dostajemy w wyniku liczbę, która nie była wcześniej zdefiniowana jako jedna ze stałych enumeracji. Przykład:

```

1 {attribute 'qualified_only'}
2 {attribute 'strict'}
3 TYPE E_Enum :
4 (
5     enum_1 := 1,
6     enum_2 := 2,
7     enum_3 := 3
8 );
9 END_TYPE

```

Został zdefiniowany typ danych o nazwie **E\_Enum**. Jako stałe enumeracji zostały wykorzystane wartości 1,2 oraz 3. W kodzie programu powstał zapis:

```

1 PROGRAM MAIN
2 VAR
3     enum1 :E_Enum;
4 END_VAR
5
6 enum1:=E_Enum.enum_3+2;

```

w wyniku takiego zapisu, zmiennej **ENUM**, zostałyby przypisana wartość 5, która nie była wcześniej zdefiniowana jako jedna ze stałych enumeracji. Dzięki atrybutowi **Strict**, kompilator od razu zgłosi błąd.

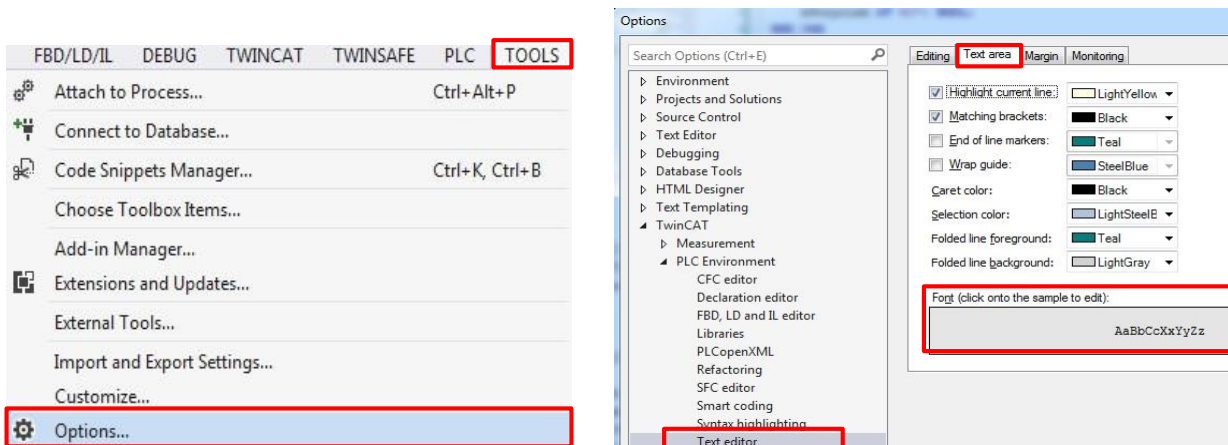
✖ 19	Arithmetics not allowed on strict ENUM type 'E_Enum'	MAIN.TcPOU	1	1
✖ 20	'(E_Enum.enum_3 + INT#2)' is not a valid value for strict ENUM type 'E_Enum'	MAIN.TcPOU	1	1

## 3.9 Zmiana ustawień środowiska

### 3.9.1 Zmiana czcionki środowiska

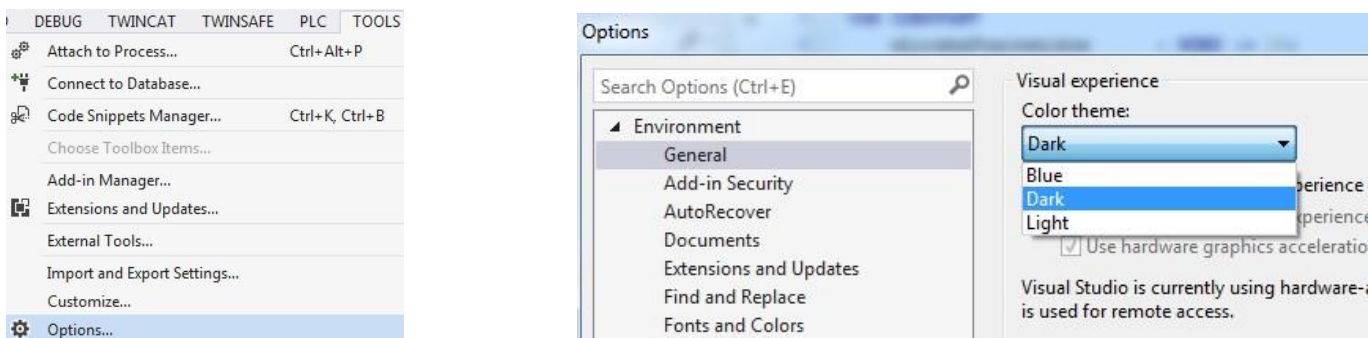
Aby zmienić czcionkę dla całego środowiska, należy wejść w **TOOLS -> Options**. Następnie należy odszukać ustawienia **TwinCAT -> PLC Enviroment -> Text Editor** i w zakładce **Text Area**, kliknąć na pole Font.





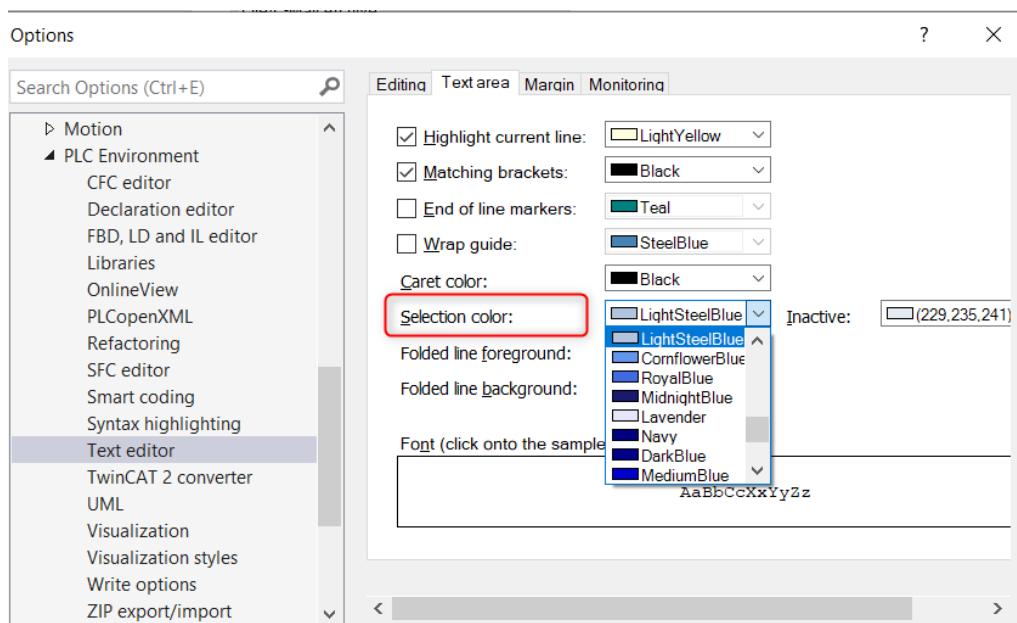
### 3.9.2 Zmiana stylu środowiska

Aby zmienić styl środowiska należy wybrać **TOOLS -> Options**. Następnie w ustawieniach **Environment -> General** z rozwijanego menu można wybrać kolor środowiska: Niebieski, Ciemny, Jasny.



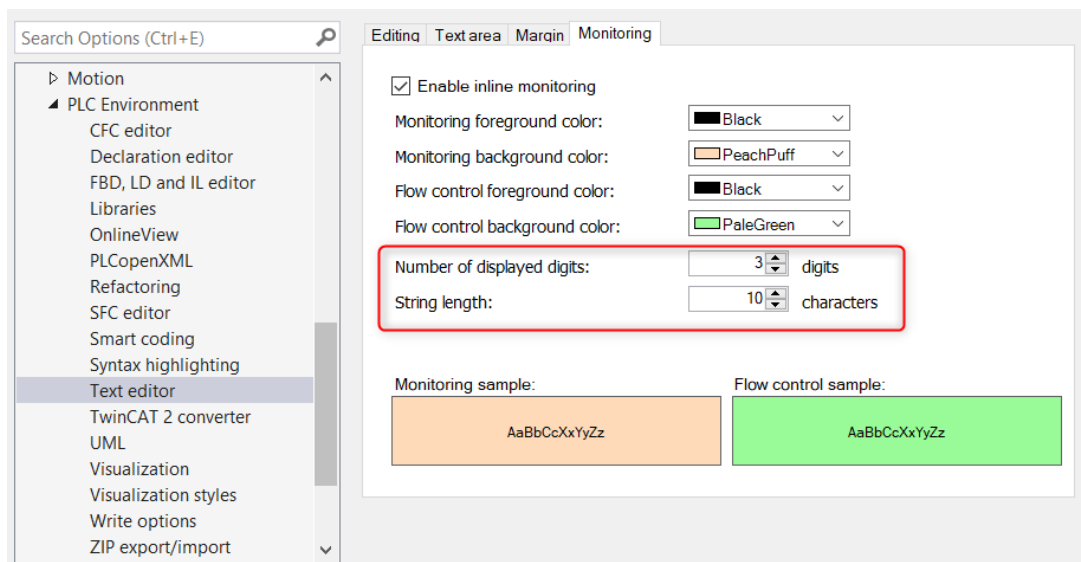
### 3.9.3 Zmiana domyślnego koloru zaznaczania

Aby zmienić domyślny kolor podświetlenia zaznaczenia, należy w **Tools -> Options** przejść do Zakładki **TwinCAT -> PLC Environment -> Text Editor**, tam wybrać zakładkę **Text area** i zmienić kolor w polu **Selection color**.

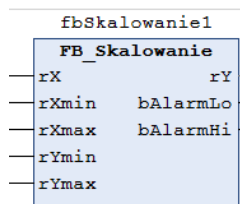


### 3.9.4 Zmiana długości wyświetlanych zmiennych w podglądzie online

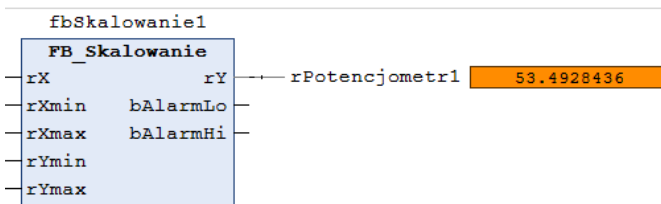
Aby zmienić domyślną długość wyświetlanych zmiennych typu REAL i STRING w podglądzie online np. w języku CFC, należy w **Tools -> Options** przejść do Zakładki **TwinCAT -> PLC Enviroment -> Text Editor**, tam wybrać zakładkę **Monitoring** i zmienić pola **Number of displayed digits** i **String length**.



Widok przed:



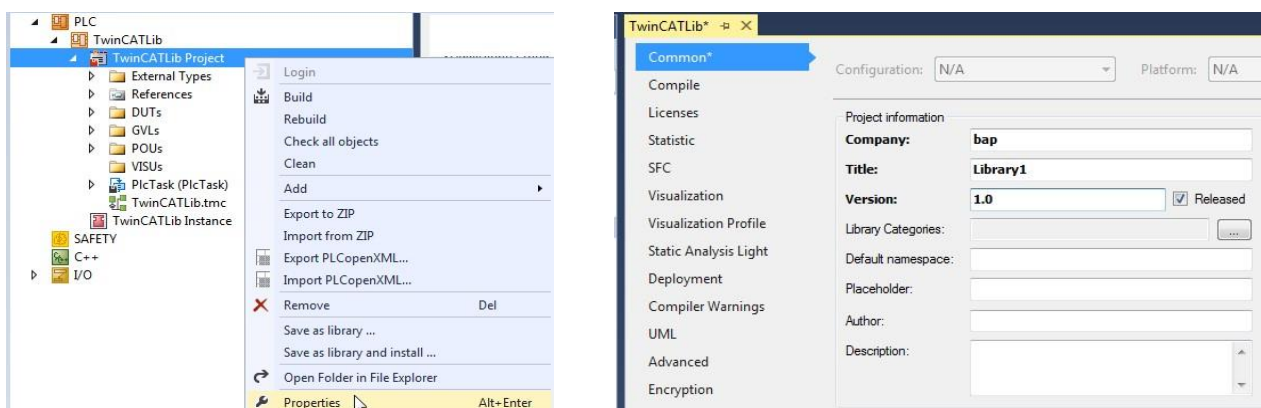
Widok po zwiększeniu:



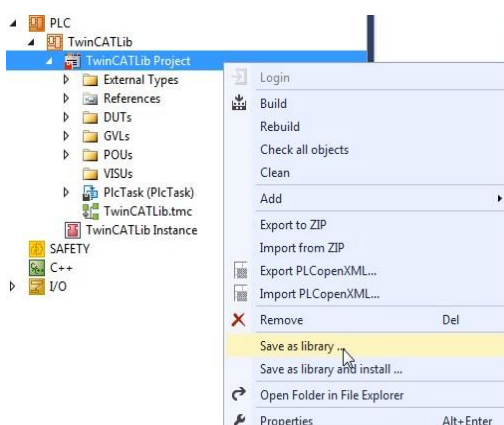
## 4 Biblioteki

### 4.1 Zapisywanie projektu w formie biblioteki

Aby zapisać projekt w formie biblioteki należy najpierw ustawić pewne właściwości projektu. W tym celu należy kliknąć PPM na nazwie projektu i wybrać **Properties**. W zakładce Common należy uzupełnić pola: **Company**, **Title**, **Version**.

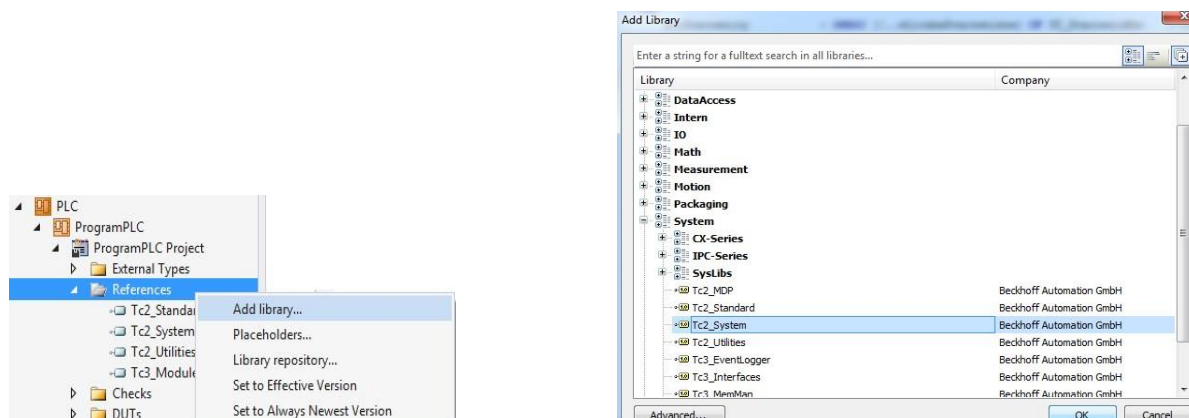


Po uzupełnieniu właściwości projektu można kliknąć na jego nazwie PPM i wybrać opcję **Save as library...** Następnie należy wybrać miejsce docelowe na dysku i zapisać bibliotekę. Plik będzie miał rozszerzenie **.library**.



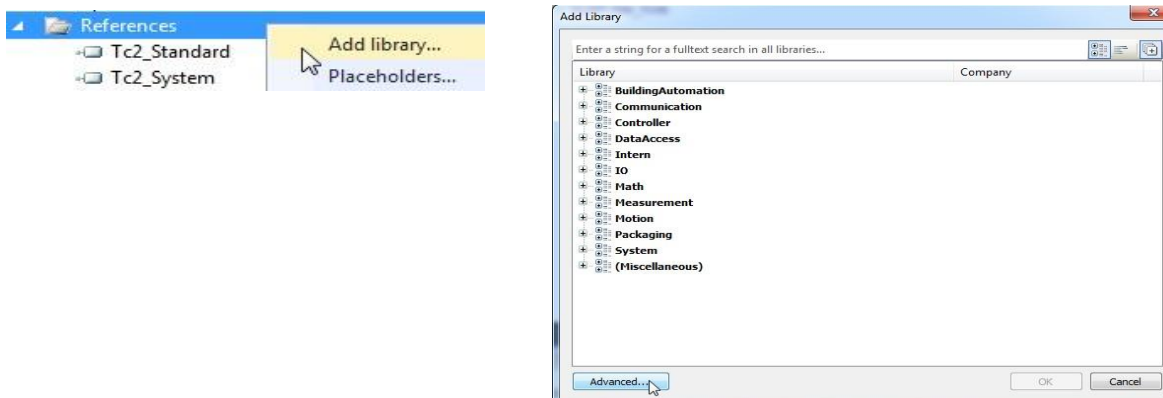
### 4.2 Dodawanie bibliotek systemowych do projektu

Aby dodać do projektu bibliotekę, z puli bibliotek dostępnych po zainstalowaniu TwinCAT 3, należy w projekcie PLC kliknąć PPM na **References** -> **Add library...** Następnie wybieramy bibliotekę z listy i klikamy OK.

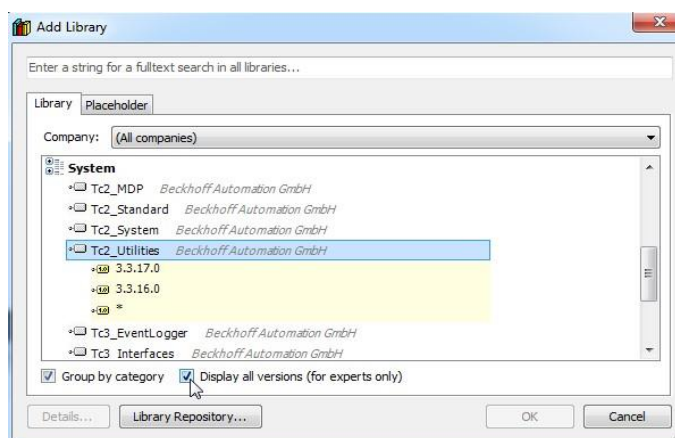


### 4.3 Korzystanie ze specyficznej wersji biblioteki

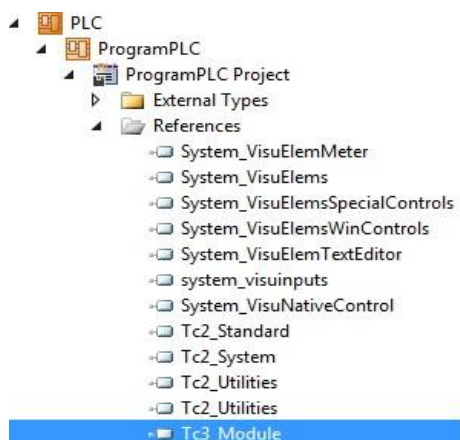
Niektóre biblioteki mają więcej niż jedną wersję i do projektu można dodać kilka wersji jednej biblioteki. Aby to zrobić należy PPM kliknąć na References i wybrać **Add library...** W oknie dialogowym które się pojawi należy wybrać przycisk **Advanced...**



Następnie, w kolejnym oknie, należy zaznaczyć opcję **Display all versions**. W przykładzie zostaną dodane dwie wersje biblioteki Tc2\_Uilities (dostępne są 3.3.17.0, 3.3.16.0, \* oznacza korzystanie z najnowszej wersji). Aby dodać bibliotekę, należy zaznaczyć ją na liście i wybrać OK.



Teraz w Referencjach znajdują się 2 biblioteki o takiej samej nazwie:



```
1 PROGRAM MAIN
2 VAR
3   ntGetTime : NT_GETTIME;
4 END
5
```

ambiguous use of name 'NT\_GETTIME'

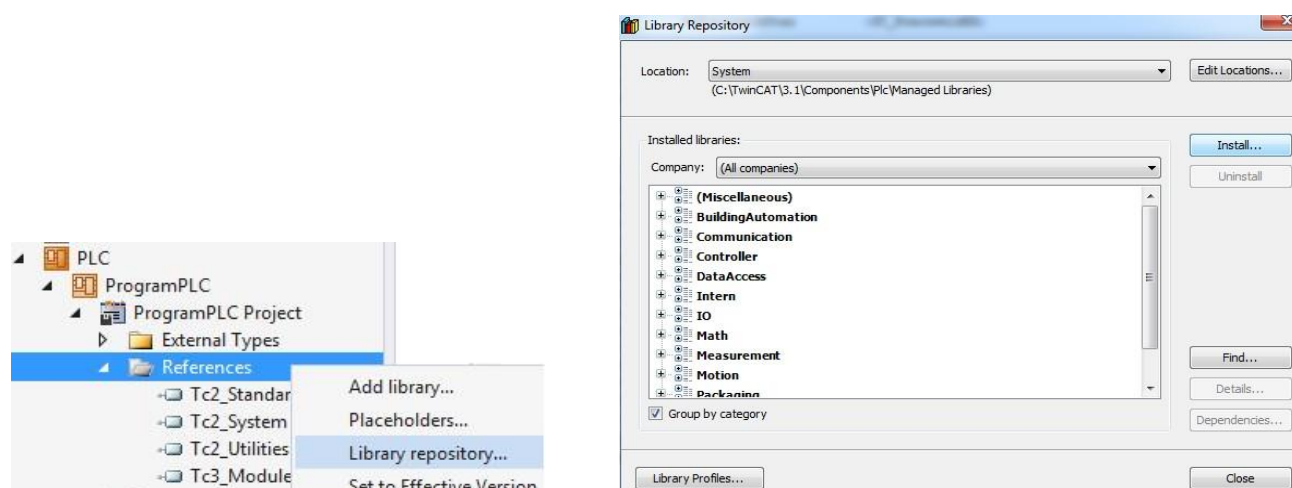
Gdy teraz w projekcie zostanie zadeklarowany blok funkcyjny z biblioteki Tc2\_Uilities to po przebudowaniu projektu pojawi się błąd. Błąd ten informuje, że kompilator nie wie z jakiej wersji biblioteki ma korzystać. W celu rozwiązania tego problemu, należy skorzystać z właściwości Namespace.

Aby przypisać bibliotece odpowiedni **Namespace**, należy wybrać ją z listy Referencji i w **Properties** biblioteki uzupełnić pole **Namespace**. W przykładzie do wersji biblioteki 3.3.16.0 została przypisana nazwa **Older**, a do wersji 3.3.17.0 nazwa **Newer**. Teraz, deklarując blok z biblioteki Tc2\_Utilities, należy dodatkowo określić z której wersji biblioteki będziemy korzystać. Określa się to używając właśnie odpowiedniego **Namespace**.



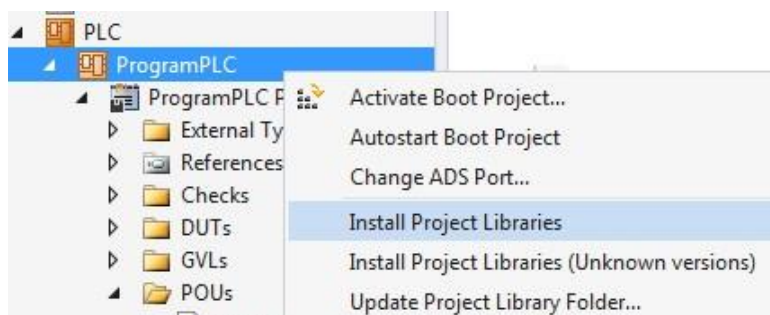
## 4.4 Instalowanie własnych bibliotek

Aby dodać do projektu bibliotekę która nie jest domyślną biblioteką dla TwinCAT, należy ją najpierw zainstalować. W tym celu należy kliknąć PPM na **References -> Library repository...** W oknie dialogowym które się pojawi wybieramy przycisk **Install** i dodajemy z dysku plik z biblioteką z rozszerzeniem **.library**. Plik powinien dodać się w kategorii **(Miscellaneous)**. Po instalacji bibliotkę można dodać do projektu poprzez opcję **Add library**.



## 4.5 Przekazywanie projektów z bibliotekami własnymi

Przy pokazywaniu projektów w których znajdują się biblioteki nie będące domyślnymi bibliotekami dla TwinCAT, po otwarciu projektu należy biblioteki te doinstalować. Jeśli projekt został przekazany w odpowiedniej formie wystarczy kliknąć PPM na nazwie projektu PLC i wybrać opcję **Install Project Libraries**.

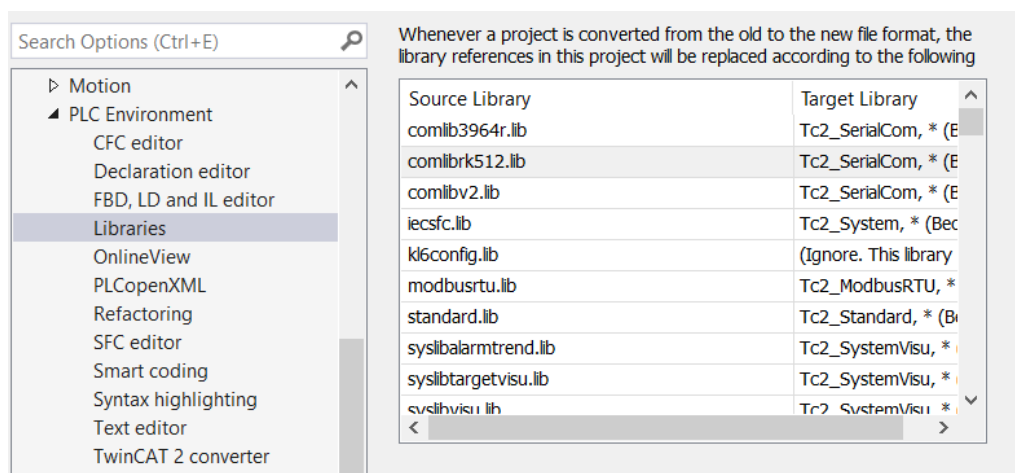




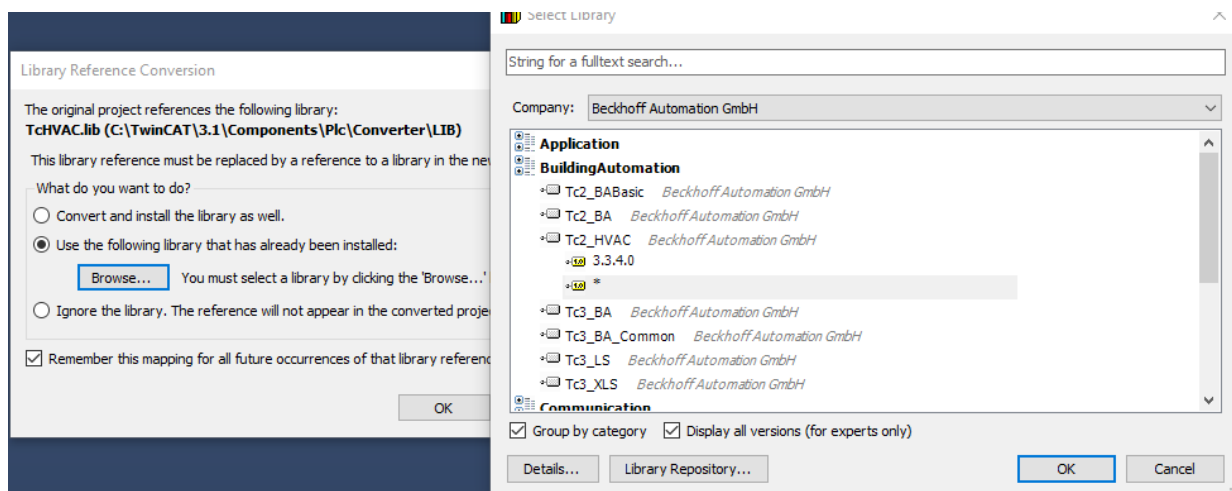
## 4.6 Kopiowanie bibliotek z TC2 do TC3

Czasami po konwersji projektu z TwinCAT 2 do TwinCAT 3 może wystąpić problem z niektórymi blokami funkcyjnymi znajdującymi się w importowanych bibliotekach. Aby go naprawić należy wykonać następujące kroki:

- Skopiować brakujące biblioteki z lokalizacji TC2: **C:\TwinCAT\Plc** do lokalizacji **C:\TwinCAT\3.1\Components\Plc\Converter\Lib**
- Usunąć bieżące mapowanie konwersji biblioteki w opcjach Visual Studio. **Tools -> Options -> TwinCat -> PLC Environment -> Libraries.**



- Po utworzeniu projektu nowego solution i zaimportowaniu projektu z TC2 wskazać na jaką bibliotekę w TC3 ma być użyta konwersja ( wraz z zaznaczonym checkbox „Remember this mapping for...” ).



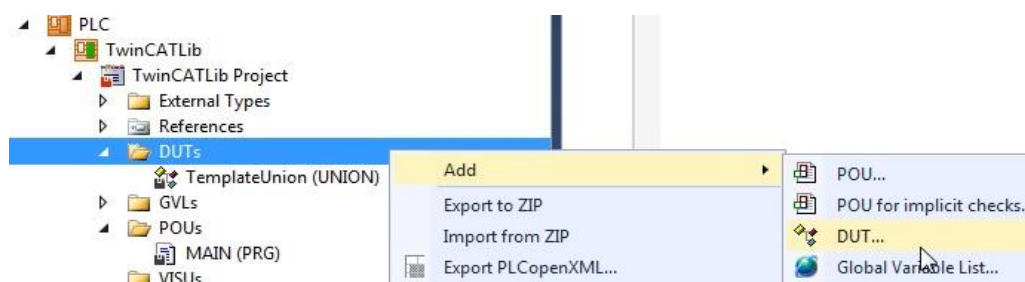
## 5 Nowe typy danych w TwinCAT 3

### 5.1 Union

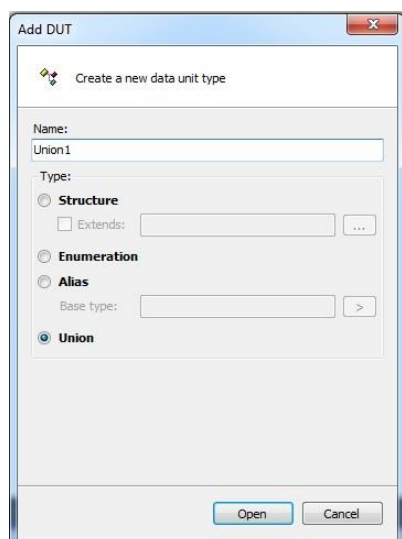
Tworząc Union możemy w jego pola wpisywać różne typy zmiennych (podobnie jak w strukturach). W Unii poszczególne pola są nakładane na ten sam obszar pamięci, a rozmiar Unii jest równy rozmiarowi największej zmiennej. Zatem zmieniając wartość jednego pola, wpływamy od razu na wartości pozostałych pól. **UWAGA!**

Nałożenie na siebie dwóch różnych typów zmiennych nie zawsze działa tak samo jak konwersja typów!!! (np. nałożenie na siebie zmiennej typu REAL i typu STRING, działa inaczej niż konwersja tych typów).

Uniony tworzy się w folderze DUTs. Należy PPM kliknąć na folder DUTs i wybrać **Add -> DUT...**



W polu dialogowym które się pojawi, należy uzupełnić pole **Name** i zaznaczyć opcję **Union**. Otwarty zostanie pusty szablon Unii, który można uzupełnić.



```
1  TYPE Union1 :
2  UNION
3  END_UNION
4  END_TYPE
5
6
```

```
1  TYPE Union1 :
2  UNION
3      wVariable   : WORD;
4      iVariable   : INT;
5      rVariable   : REAL;
6  END_UNION
7  END_TYPE
```

Aby wykorzystać Union w programie, należy go najpierw zadeklarować. Odwołanie do pola Unii odbywa się poprzez podanie nazwy Uniona, a po kropce wybraniu odpowiedniego pola.

```

1 PROGRAM MAIN
2 VAR
3   uTemplate : U_Template;
4 END_VAR
5
1 uTemplate.
  iVariable
  rVariable
  wVariable

```

Expression	Type	Value
uTemplate	U_Template	
wVariable	WORD	32600
iVariable	INT	32600
rVariable	REAL	4.568233E-41
iVar	INT	32600
wVar1	WORD	32600

```

1 iVar 32600 :=uTemplate.iVariable 32600;
2 wVar1 32600 :=uTemplate.wVariable 32600; RETURN

```

## 5.2 BIT

Typ danych BIT może być używany jako indywidualna zmienna jedynie wewnątrz struktur. Dopuszczalne wartości tej zmiennej to TRUE(1) oraz FALSE(0). Element typu BIT wymaga 1 bit pamięci i pozwala na dostęp do konkretnego bitu struktury poprzez jego nazwę. Kolejno zadeklarowane BITy są łączone w bajty co pozwala na optymalizację zużycia pamięci (w porównaniu do używania zmiennych typu BOOL).

Dostęp do danej typu BIT zajmuje znacząco więcej czasu niż w przypadku zmiennej BOOL. Typ BIT powinien być używany jedynie w aplikacjach wymagających dokładnej specyfikacji formatu danych.

```

ST_Bit
1 TYPE ST_Bit :
2 STRUCT
3   bBit1 : BIT;
4   bBit2 : BIT;
5   bBit3 : BIT;
6 END_STRUCT
7 END_TYPE

```

```

MAIN
1 PROGRAM MAIN
2 VAR
3   stBit : ST_Bit;
4   bZmienna : BOOL;
5 END_VAR
6
1 bZmienna:=stBit.bBit1;

```

## 5.3 LTIME

Zmienna typu LTIME jest zmienną 64-bitową (zmienna TIME – 32 bit). Rozdzielczość tej zmiennej określa się w nanosekundach (typ TIME – milisekundy).

Przykładowa deklaracja tej zmiennej:

**LTIME1 := LTIME#1000d15h23m12s34ms2us44ns;**

Gdzie: us – oznacza mikrosekundy, ns – oznacza nanosekundy

```

1 PROGRAM MAIN
2 VAR
3   lTime1 : LTIME;
4 END_VAR
5
1 lTime1:=LTIME#1000D15H23M12S34MS2US44NS;

```

Expression	Type	Value	Prepared value	Address
lTime1	LTIME	LTIME#1000d15h23m12s34ms2us44ns		
lTime1		LTIME#1000d15h23m12s34ms2us44ns		
lTime1		LTIME#1000D15H23M12S34MS2US44NS		



## 5.4 REFERENCE

REFERENCE jest aliasem dla obiektu i może być nadpisywany bądź odczytywany za pomocą identyfikatorów. Przykład deklaracji:

```
1 PROGRAM MAIN
2 VAR
3     ref_int : REFERENCE TO
4     INT;
5     iVar1   : INT;
6     iVar2   : INT;
7 END_VAR
```

ref\_int może być teraz używany jako alias dla zmiennych typu INT

```
1 ref_int REF= iVar1; (* ref_int odnosi sie teraz do zmiennej iVar1 *)
2 ref_int := 12; (* iVar1 ma teraz wartosc 12 *)
3 iVar2 := ref_int * 2; (* iVar2 ma teraz wartosc 24 *)
4 ref_int REF= iVar2; (* ref_int odnosi sie teraz do zmiennej iVar2 *)
5 ref_int := iVar1 / 2; (* iVar2 nma teraz wartosc 6 *)
6 ref_int REF= 0; (* inicjalizacja elementu ref_int*)
```

## 5.5 Zmiana typu wyliczeniowego enumeracji

Zmiany dotyczące typu wyliczeniowego dotyczą możliwości określenia jakiego typu ma być stała enumeracji. Domyślnie jest to typ INT. Jeśli chcemy aby stała enumeracji była np. typu DINT, składania wygląda następująco:

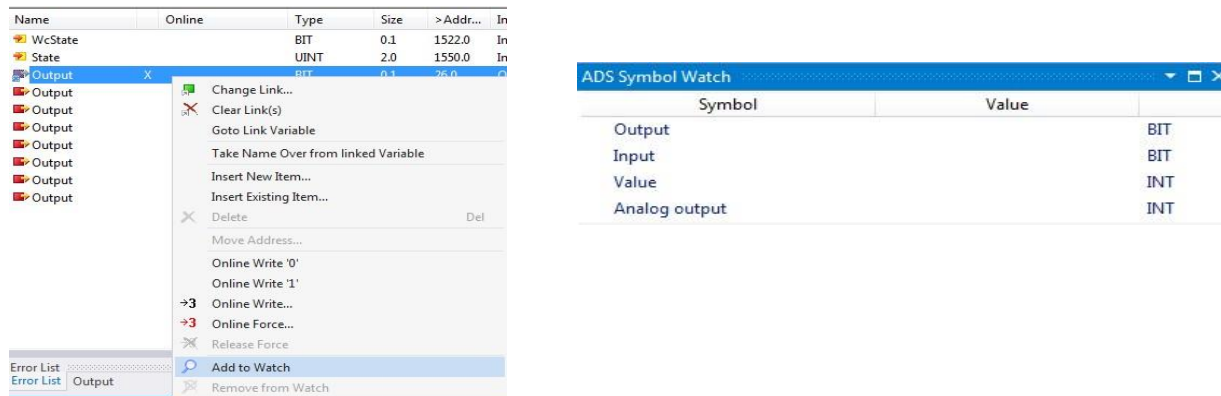
```
1 TYPE E_DINT1 :
2 (
3     eYellow,
4     eBlue,
5     eGreen :=16#8000
6 ) DINT;
7 END_TYPE
```

typ stałej enumeracji.

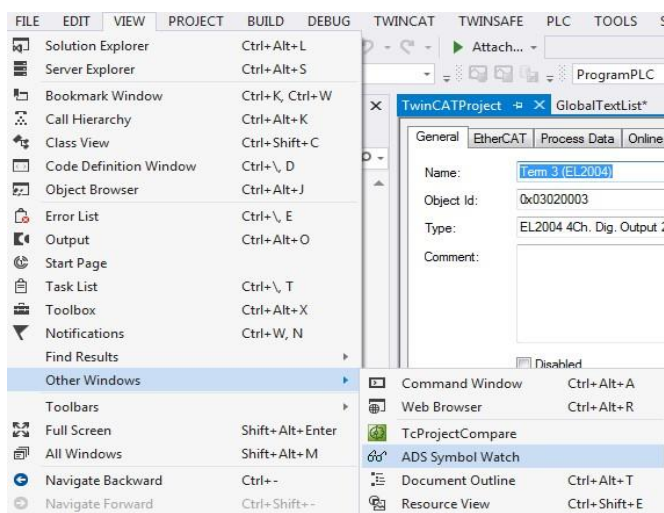
## 6 Tips & tricks

### 6.1 Watchlist

Aby dodać zmienne do swojej Watchlisty należy kliknąć na wybraną zmienną PPM a następnie wybrać **Add To Watch**. Pojawi się okno **ADS Symbol Watch** gdzie będą widoczne wybrane zmienne. Po zapisaniu projektu, **Watchlist** również zostanie zapisany.

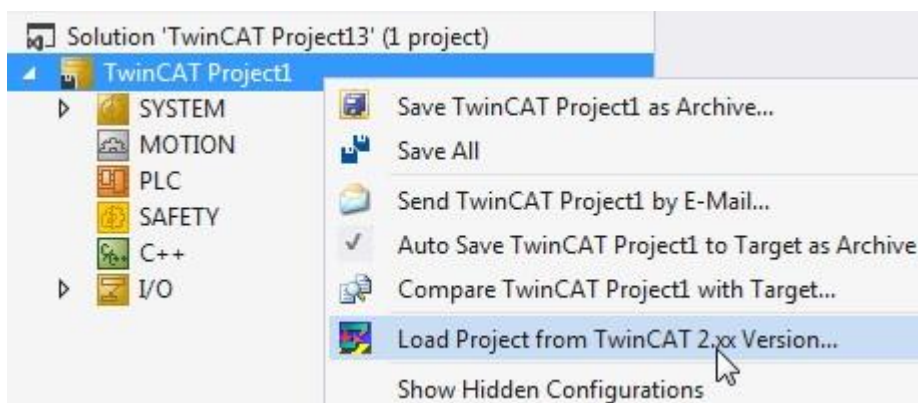


Aby wywołać Watchlist, należy wybrać **View -> Other Windows -> ADS Symbol Watch**:



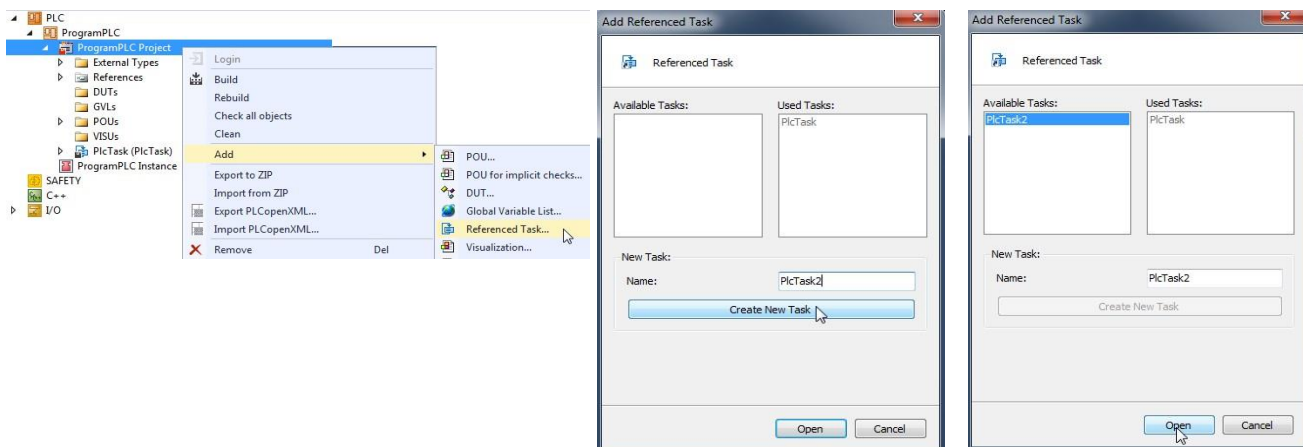
### 6.2 Otwieranie aplikacji stworzonej w TwinCAT 2 w TwinCAT 3

Aby otworzyć w TC3 aplikację stworzoną w TC2, należy kliknąć PPM na nazwę projektu w TC3 i wybrać opcję **Load Project from TwinCAT 2.xx Version**. W oknie które się pojawi należy wskazać plik z rozszerzeniem .tsm czyli plik konfiguracyjny z TC2 zawierający ścieżkę do programu.



## 6.3 Dodawanie nowych tasków

Podczas dodawania tasków, należy zwracać uwagę na priorytety. Dobrą praktyką podczas dodawania tasków, jest robienie tego poprzez kliknięcie PPM na nazwie projektu PLC i wybrać **Add -> Referenced Task...** W oknie dialogowym, w polu **Name**, należy wpisać nazwę nowego tasku i wybrać przycisk **Create New Task**. Dodany task pojawi się w oknie **Available Task**, należy go w tym oknie zaznaczyć i wybrać **Open**.



Ustawienia priorytetów można sprawdzić, wybierając z drzewa projektu **Real-Time**, a następnie zakładkę **Priorities**. Aby dokonywać zmian ustawień tasków, należy wybrać odpowiedni task w drzewie projektu, w zakładce **Tasks**.

