BECKHOFF

1 second UPS

Poziom trudności: łatwy

Wersja dokumentacji: 1.8

Aktualizacja: 29.05.2017

Beckhoff Automation Sp. z o. o.

Spis treści

1. Wprowadzenie 3
2. Implementacja funkcjonalności 1 second UPS w TwinCAT 2 4
2.1. Domyślne wywołanie bloku FB_S_UPS 4
2.2. Domyślne wywołanie bloku FB_S_UPS_CX80xx, FB_S_UPS_CX9020_U900, FB_S_UPS_CX51x0, FB_S_UPS_CB3011 4
2.3. Wywołanie bloku FB_S_UPS z własnymi ustawieniami 5
2.4. Wywołanie bloku FB_S_UPS_UPS_CX80xx z własnymi ustawieniami 6
3. Implementacja funkcjonalności 1 second UPS w TwinCAT 3 7
3.4. Domyślne wywołanie bloku FB_S_UPS 7
3.2. Domyślne wywołanie bloku FB_S_UPS_CX9020_U900, FB_S_UPS_CX51x0, FB_S_UPS_CB3011 7
3.3. Wywołanie bloku FB_S_UPS z własnymi ustawieniami 8
3.4. Wywołanie bloku FB_S_UPS_UPS_CX9020_U900 z własnymi ustawieniami 9
4. Zmienne PERSISTENT 10

1. Wprowadzenie

1 second UPS jest to funkcjonalność umożliwiająca, po wykryciu zaniku zasilania, zapisanie danych na karcie (np. Compact Flash). Zmienne te muszą być oznaczone jako PERSISTENT.

1 second UPS składa się z wbudowanego prostego UPS oraz bloku funkcyjnego *FB_S_UPS* wywołanego w programie TwinCAT.

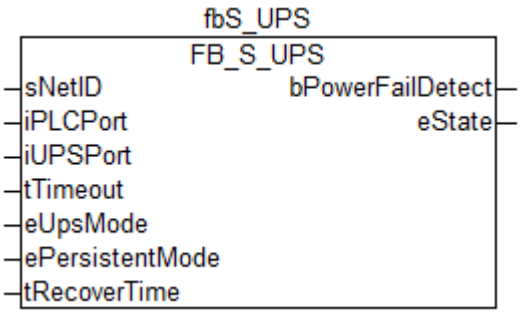
1 second UPS jest dostępny dla następujących komputerów przemysłowych oraz Embedded PC:

Sterownik	**1 second UPS**	**Blok Funkcyjny**	**Biblioteka**	
		TC2 & TC3	**TC2**	**TC3**
CX50x0	Wbudowany	FB_S_UPS	TcSUPS.lib	Tc2_SUPS
CX51x0	Wbudowany	FB_S_UPS_CX51x0	TcSUPS_CX51x0.lib	
CX9020	Opcja	FB_S_UPS_CX9020_U900	TcSUPS_CX9020_U900.lib	
CP66xx-0020 **CP26xx-0000**	Opcja	FB_S_UPS_CB3011	TcSUPS_CB3011.lib	
C6915-0000	Opcja	FB_S_UPS	TcSUPS.lib	
CP62xx-0020				
CP77xx-0030				
CX80xx	Wbudowany	FB_S_UPS_CX80xx	TcSystemCX80xx.lib	Brak

Uwaga!! W przypadku TwinCAT 3 należy do projektu dodać dodatkowo bibliotekę Tc2_Utilities.library

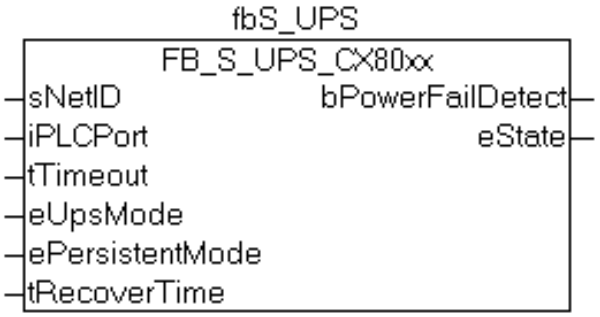
2. Implementacja funkcjonalności 1 second UPS w TwinCAT 2

2.1. Domyślne wywołanie bloku FB_S_UPS



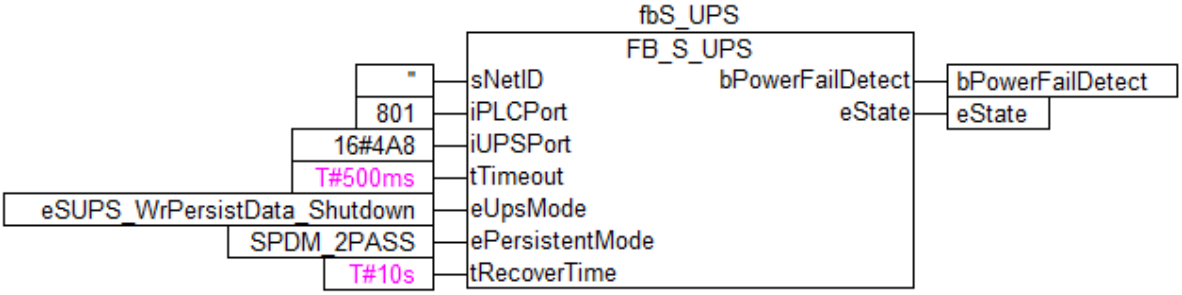
Uwaga!! Wystarczy wywołać blok funkcyjny w programie i nadać mu nazwę, nie ma potrzeby podpinania wejść i wyjść – przyjęte zostaną parametry domyślne (podane w rozdziale 2.3). Przykładowe wywołanie pokazane jest na ilustracji powyżej.

2.2. Domyślne wywołanie bloku FB_S_UPS_CX80xx, FB_S_UPS_CX9020_U900, FB_S_UPS_CX51x0, FB_S_UPS_CB3011



Uwaga!! Wystarczy wywołać blok funkcyjny w programie i nadać mu nazwę, nie ma potrzeby podpinania wejść i wyjść – będzie działał z parametrami domyślnymi (podane w rozdziale 2.4). Przykładowe wywołanie pokazane jest na ilustracji powyżej.

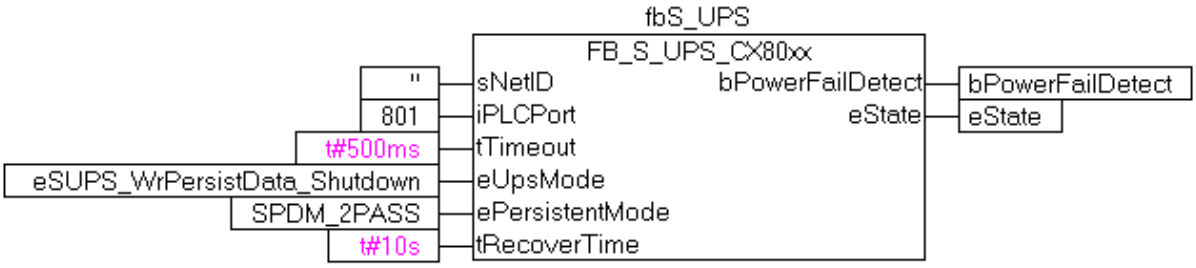
2.3.Wywołanie bloku FB_S_UPS z własnymi ustawieniami



Opis poszczególnych wejść i wyjść:

- ***sNetID*** – adres AmsNetId sterownika w formie zmiennej typu *STRING,* domyślnie zapisujemy lokalnie więc pozostawiamy pusty wpis,,**''**,,
- ***iPLCPort*** – numer portu, możemy wpisać wartość typu INT lub wartość typu *ENUM*:
  RunTime 1: ***AMSPORT_R0_PLC_RTS1*** lub ***801*** – wartość domyślna,
  RunTime 2: *AMSPORT_R0_PLC_RTS2* lub *811,*
  RunTime 3: *AMSPORT_R0_PLC_RTS3* lub *821,*
  RunTime 4: *AMSPORT_R0_PLC_RTS4* lub *831.*
- ***iUPSPort*** – numer portu z którego będziemy odczytywali stan zasilania – domyślnie ***16#4A8***,
- ***tTimeout*** – czas oczekiwania na wykonanie szybkiego zamknięcia systemu w formie zmiennej typu TIME (np.:*T#750ms*), domyślnie **DEFAULT_ADS_TIMEOUT** lub **T#5s**,
- ***eUPSMode*** – opisuje tryb pracy bloczka *FB_S_UPS*, wartość domyślna **eSUPS_WrPersistData_Shutdown.** Tryby pracy:
  - ***eSUPS_WrPersistData_Shutdown*** – zapisuje zmienne typu PERSISTENT, następnie wykonuje szybkie zamknięcie systemu,
  - *eSUPS_WrPersistData_NoShutdown* – zapisuje tylko zmienne typu PERSISTENT bez wykonania szybkiego zamknięcia system,
  - *eSUPS_ImmediateShutdown* – wykonuje tylko szybkie zamknięcie system bez zapisywania zmiennych *PERSISTENT*,
  - *eSUPS_CheckPowerStatus* – sprawdza tylko status zasilania bez zapisywania zmiennych PERSISTENT oraz bez szybkiego zamykania systemu podczas awarii zasilania.
- ***ePersistentMode*** – tryb pracy zapisu zmiennych *PERSISTENT*. Domyślnie ustawione na ***SPDM_2PASS*** pozwala na zapisanie danych nawet jeżeli może to wydłużyć czas wykonania się programu,
- ***tRecoverTime*** - czas, który blok funkcyjny będzie czekał po awarii zasilania do stanu *PowerOK*. Zmienna musi być trochę większa od czasu podtrzymania zasilania, domyślnie **T#10s**,
- ***bPowerFailDetect*** – przyjmuje wartość *TRUE* podczas awarii zasilania, a kiedy zasilanie jest poprawne wyjście przyjmuje wartość *FALSE,*
- ***eState*** – wewnętrzny stan bloku funkcyjnego w przypadku, gdy zasilanie jest poprawne zmienna przyjmuje wartość *eSUPS_PowerOK.*

2.4.Wywołanie bloku FB_S_UPS_UPS_CX80xx z własnymi ustawieniami



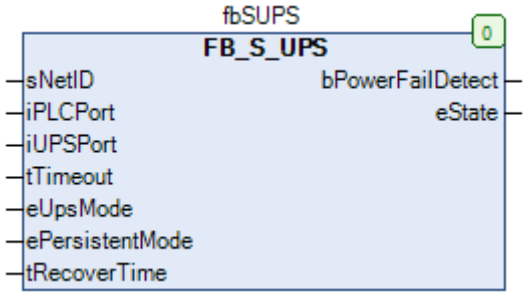
Opis poszczególnych wejść i wyjść:

- ***sNetID*** – adres AmsNetId sterownika w formie zmiennej typu *STRING,* domyślnie zapisujemy lokalnie więc pozostawiamy pusty wpis,,**''**,,
- ***iPLCPort*** – numer portu, możemy wpisać wartość typu INT lub wartość typu *ENUM*:

  RunTime 1: ***AMSPORT_R0_PLC_RTS1*** lub ***801*** – wartość domyślna,
  RunTime 2: *AMSPORT_R0_PLC_RTS2* lub *811,*
  RunTime 3: *AMSPORT_R0_PLC_RTS3* lub *821,*
  RunTime 4: *AMSPORT_R0_PLC_RTS4* lub *831.*
- ***tTimeout*** – czas oczekiwania na wykonanie szybkiego zamknięcia systemu w formie zmiennej typu TIME (np.:*T#750ms*), domyślnie **DEFAULT_ADS_TIMEOUT** lub **T#5s**,
- ***eUPSMode*** – opisuje tryb pracy bloczka *FB_S_UPS*, wartość domyślna **eSUPS_WrPersistData_Shutdown.** Tryby pracy:
  - ***eSUPS_WrPersistData_Shutdown*** – zapisuje zmienne typu PERSISTENT, następnie wykonuje szybkie zamknięcie systemu,
  - *eSUPS_WrPersistData_NoShutdown* – zapisuje tylko zmienne typu PERSISTENT bez wykonania szybkiego zamknięcia system,
  - *eSUPS_ImmediateShutdown* – wykonuje tylko szybkie zamknięcie system bez zapisywania zmiennych *PERSISTENT*,
  - *eSUPS_CheckPowerStatus* – sprawdza tylko status zasilania bez zapisywania zmiennych PERSISTENT oraz bez szybkiego zamykania systemu podczas awarii zasilania.
- ***ePersistentMode*** – tryb pracy zapisu zmiennych *PERSISTENT*. Domyślnie ustawione na ***SPDM_2PASS*** pozwala na zapisanie danych nawet jeżeli może to wydłużyć czas wykonania się programu,
- ***tRecoverTime*** - czas, który blok funkcyjny będzie czekał po awarii zasilania do stanu *PowerOK*. Zmienna musi być trochę większa od czasu podtrzymania zasilania, domyślnie **T#10s**,
- ***bPowerFailDetect*** – przyjmuje wartość *TRUE* podczas awarii zasilania, a kiedy zasilanie jest poprawne wyjście przyjmuje wartość *FALSE,*
- ***eState*** – wewnętrzny stan bloku funkcyjnego w przypadku, gdy zasilanie jest poprawne zmienna przyjmuje wartość *eSUPS_PowerOK.*

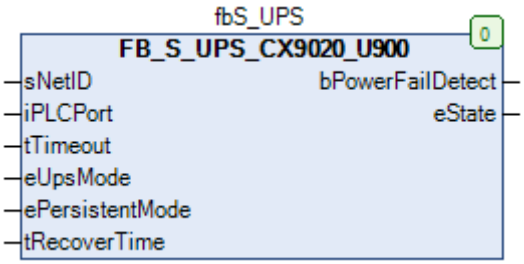
3. Implementacja funkcjonalności 1 second UPS w TwinCAT 3

3.4.Domyślne wywołanie bloku FB_S_UPS



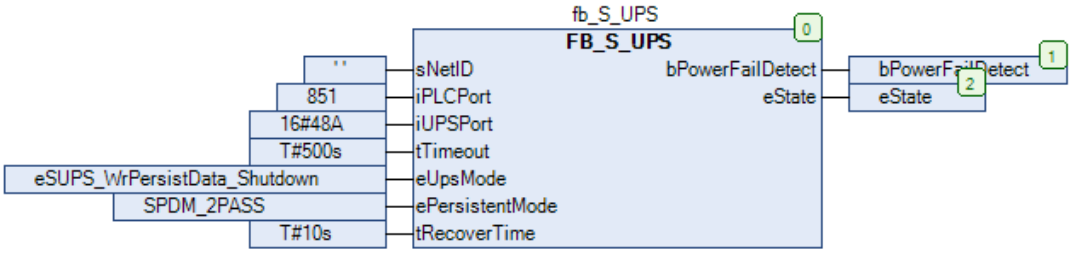
Uwaga!! Wystarczy wywołać blok funkcyjny w programie i nadać mu nazwę, nie ma potrzeby podpinania wejść i wyjść – będzie działał z parametrami domyślnymi (podane w rozdziale 3.3). Przykładowe wywołanie pokazane jest na ilustracji powyżej.

3.2.Domyślne wywołanie bloku FB_S_UPS_CX9020_U900, FB_S_UPS_CX51x0, FB_S_UPS_CB3011



Uwaga!! Wystarczy wywołać blok funkcyjny w programie i nadać mu nazwę, nie ma potrzeby podpinania wejść i wyjść – będzie działał z parametrami domyślnymi (podane w rozdziale 3.4). Przykładowe wywołanie pokazane jest na ilustracji powyżej.

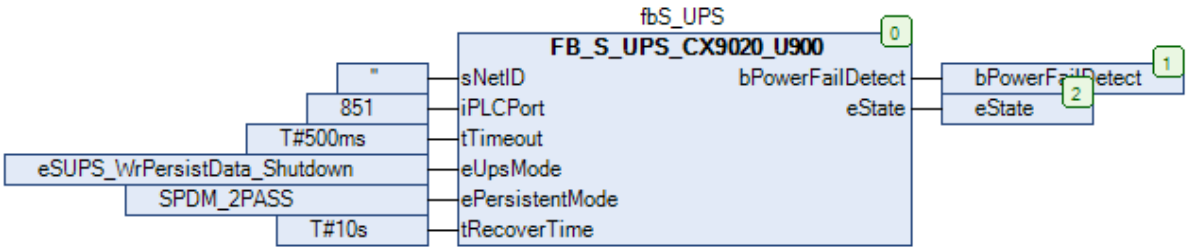
3.3.Wywołanie bloku FB_S_UPS z własnymi ustawieniami



Opis poszczególnych wejść i wyjść:

- ***sNetID*** – adres AmsNetId sterownika w formie zmiennej typu *STRING,* domyślnie zapisujemy lokalnie więc pozostawiamy pusty wpis,,**''**,,
- ***iPLCPort*** – numer portu:
  RunTime 1: ***851*** – wartość domyślna*,*
  RunTime 2: *852,*
  RunTime 3: *853,*
  RunTime 4: *854*
  RunTime n: *850 + n (max. 876)*
- ***iUPSPort*** – numer portu z którego będziemy odczytywali stan zasilania – domyślnie ***16#4A8***,
- ***tTimeout*** – czas oczekiwania na wykonanie szybkiego zamknięcia systemu w formie zmiennej typu TIME (np.:*T#750ms*), domyślnie **T#5s**,
- ***eUPSMode*** – opisuje tryb pracy bloczka *FB_S_UPS*, wartość domyślna **eSUPS_WrPersistData_Shutdown.** Tryby pracy:
  - ***eSUPS_WrPersistData_Shutdown*** – zapisuje zmienne typu PERSISTENT, następnie wykonuje szybkie zamknięcie systemu,
  - *eSUPS_WrPersistData_NoShutdown* – zapisuje tylko zmienne typu PERSISTENT bez wykonania szybkiego zamknięcia system,
  - *eSUPS_ImmediateShutdown* – wykonuje tylko szybkie zamknięcie system bez zapisywania zmiennych *PERSISTENT*,
  - *eSUPS_CheckPowerStatus* – sprawdza tylko status zasilania bez zapisywania zmiennych PERSISTENT oraz bez szybkiego zamykania systemu podczas awarii zasilania.
- ***ePersistentMode*** – tryb pracy zapisu zmiennych *PERSISTENT*. Domyślnie ustawione na ***SPDM_2PASS*** pozwala na zapisanie danych nawet jeżeli może to wydłużyć czas wykonania się programu,
- ***tRecoverTime*** - czas, który blok funkcyjny będzie czekał po awarii zasilania do stanu *PowerOK*. Wartość musi być większa od czasu podtrzymania zasilania, domyślnie **T#10s**,
- ***bPowerFailDetect*** – przyjmuje wartość *TRUE* podczas awarii zasilania, a kiedy zasilanie jest poprawne wyjście przyjmuje wartość *FALSE*
- ***eState*** – wewnętrzny stan bloku funkcyjnego w przypadku, gdy zasilanie jest poprawne zmienna przyjmuje wartość *eSUPS_PowerOK.*

3.4.Wywołanie bloku FB_S_UPS_UPS_CX9020_U900 z własnymi ustawieniami



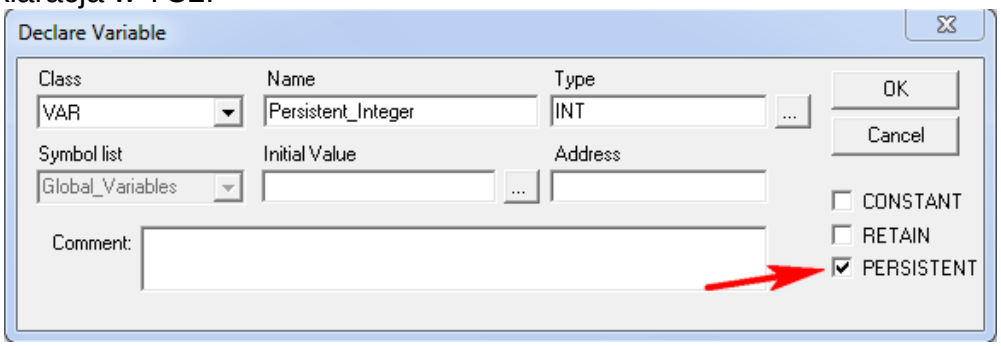
Opis poszczególnych wejść i wyjść:

- ***sNetID*** – adres AmsNetId sterownika w formie zmiennej typu *STRING,* domyślnie zapisujemy lokalnie więc pozostawiamy pusty wpis,,**''**,,
- ***iPLCPort*** – numer portu:
  RunTime 1: ***851*** – wartość domyślna,
  RunTime 2: *852,*
  RunTime 3: *853,*
  RunTime 4: *854*
  RunTime n: *850 + n (max 876)*
- ***tTimeout*** – czas oczekiwania na wykonanie szybkiego zamknięcia systemu w formie zmiennej typu TIME (np.:*T#750ms*), domyślnie **T#5s**,
- ***eUPSMode*** – opisuje tryb pracy bloczka *FB_S_UPS*, wartość domyślna **eSUPS_WrPersistData_Shutdown.** Tryby pracy:
  - ***eSUPS_WrPersistData_Shutdown*** – zapisuje zmienne typu PERSISTENT, następnie wykonuje szybkie zamknięcie systemu,
  - *eSUPS_WrPersistData_NoShutdown* – zapisuje tylko zmienne typu PERSISTENT bez wykonania szybkiego zamknięcia system,
  - *eSUPS_ImmediateShutdown* – wykonuje tylko szybkie zamknięcie system bez zapisywania zmiennych *PERSISTENT*,
  - *eSUPS_CheckPowerStatus* – sprawdza tylko status zasilania bez zapisywania zmiennych PERSISTENT oraz bez szybkiego zamykania systemu podczas awarii zasilania.
- ***ePersistentMode*** – tryb pracy zapisu zmiennych *PERSISTENT*. Domyślnie ustawione na ***SPDM_2PASS*** pozwala na zapisanie danych nawet jeżeli może to wydłużyć czas wykonania się programu,
- ***tRecoverTime*** - czas, który blok funkcyjny będzie czekał po awarii zasilania do stanu *PowerOK*. Wartość musi być większa od czasu podtrzymania zasilania, domyślnie **T#10s**,
- ***bPowerFailDetect*** – przyjmuje wartość *TRUE* podczas awarii zasilania, a kiedy zasilanie jest poprawne wyjście przyjmuje wartość *FALSE,*
- ***eState*** – wewnętrzny stan bloku funkcyjnego w przypadku, gdy zasilanie jest poprawne zmienna przyjmuje wartość *eSUPS_PowerOK.*

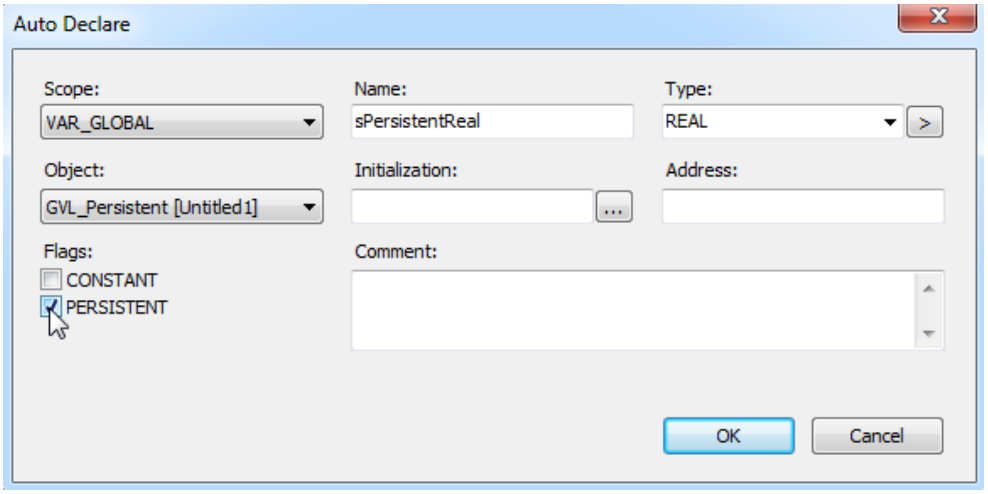
4. Zmienne PERSISTENT

Zmienne, które mają być zapisane jako *PERSISTENT* wymagają specjalnej deklaracji – muszą być umieszczone między wyrażeniami **VAR PERSISTENT i END_VAR** lub podczas automatycznej deklaracji zaznaczyć należy opcje PERSISTENT (ilustracja poniżej).

Deklaracja w TC2:

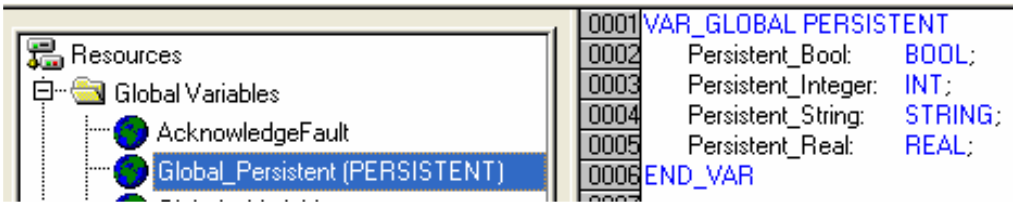


Deklaracja w TC3:

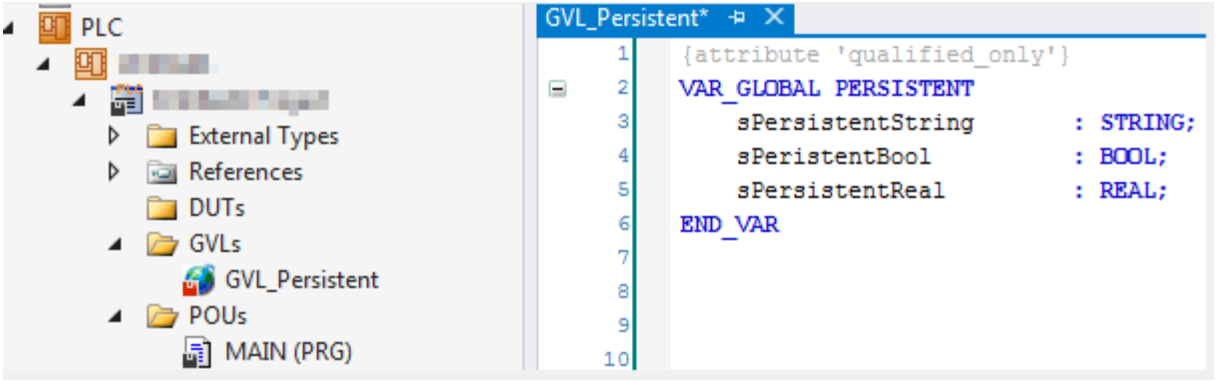


Wartość inicjalizacyjna (Initial Value – ilustracja powyżej) zmiennej *PERSISTENT* zostanie przypisana tylko w przypadku błędu odczytu zmiennych z pliku.

Zmienne *PERSISTENT* mogą być również zmiennymi globalnymi, przykład deklaracji wTC2:



Przykład deklaracji w TC3:



Dodatkowe informacje o zmiennych typu PERSISTENT znajdują się pod adresem: ftp://ftp.beckhoff.com/poland/Pomoc/Retain_Persistent/