

Komunikacja paneli Astraada HMI ze sterownikami Siemens S7-1200

WSTĘP

Panele operatorskie Astraada HMI Panel mają możliwość wymiany danych procesowych ze sterownikami Siemens S7 w sieci Ethernet. Pozwala na to odpowiedni driver komunikacyjny wbudowany w oprogramowanie narzędziowe Astraada HMI CFG. W celu uruchomienia komunikacji konieczna jest odpowiednia konfiguracja drivera po stronie panelu i sterownika.

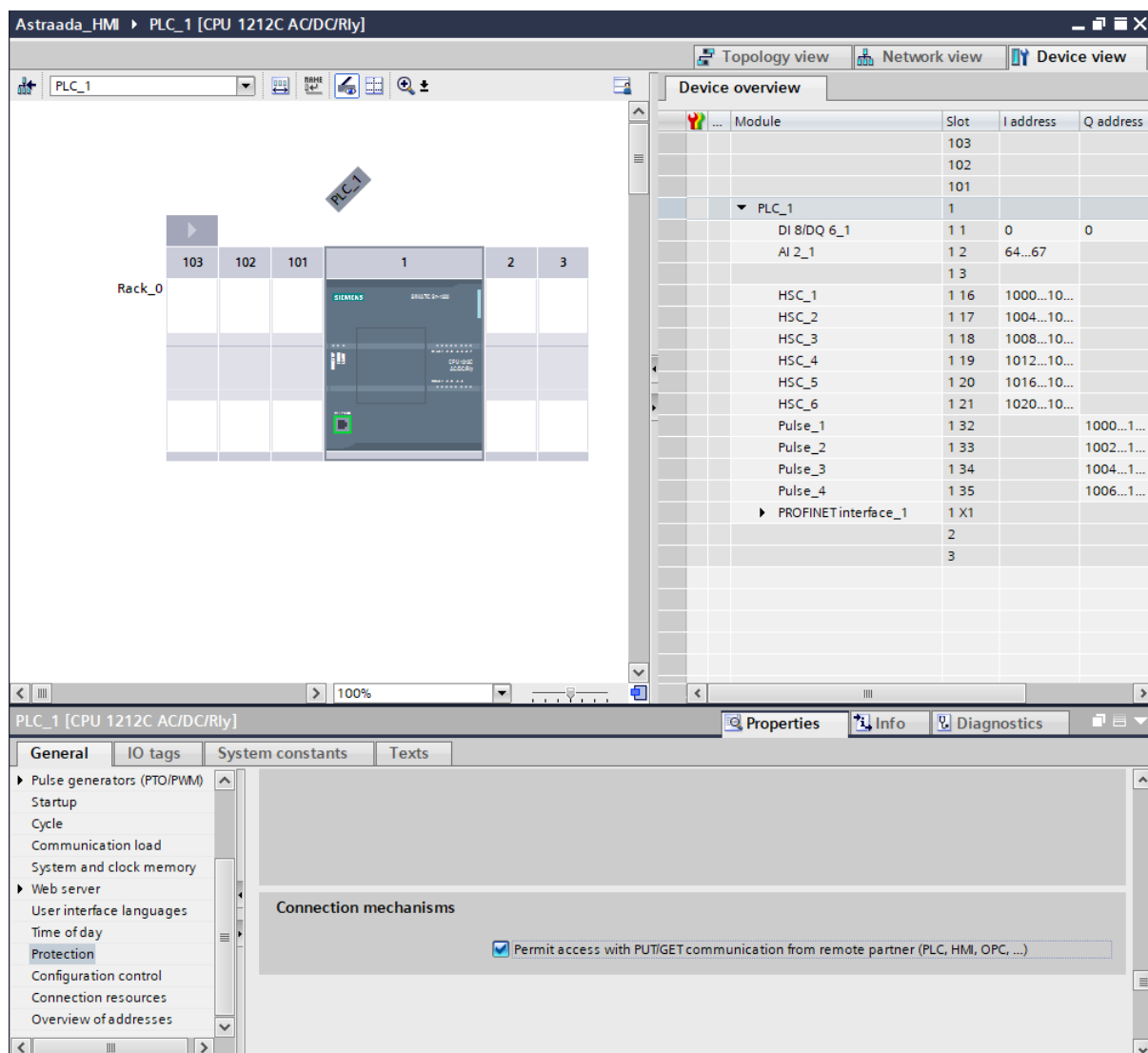
W instrukcji opisano zagadnienia związane z:

- Konfiguracją panelu Astraada HMI i sterownika Siemens S7-1200
- Adresowaniem zmiennych
- Importem własnych typów danych
- Importem zmiennych zorganizowanych w Data Blockach

Uwaga – akapity poświęcone importowaniu zmiennych w Data Blockach dotyczą tylko wersji oprogramowania Astraada HMI CFG 4.0.x.x, obsłudze importu zmiennych w starszych wersjach oprogramowania poświęcona jest osobna instrukcja dostępna w [zakładce Wsparcie na stronie internetowej firmy Astor](#).

KONFIGURACJA STEROWNIKA

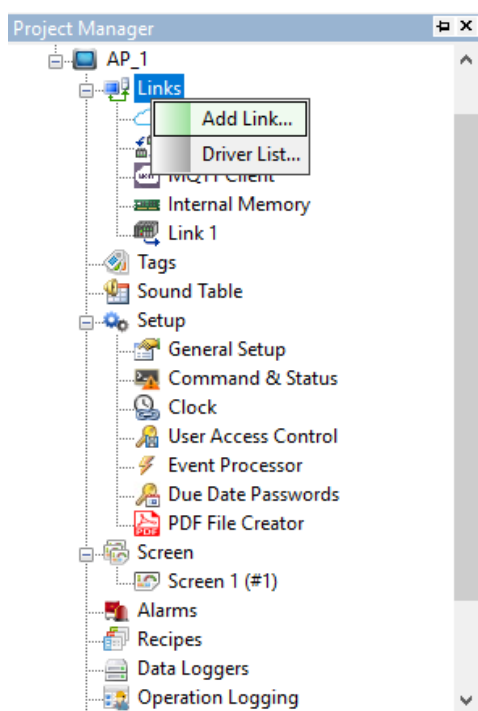
Po podstawowej konfiguracji sterownika w oprogramowaniu TIA Portal należy przejść do zakładki Properties w kontekście określonego urządzenia, a następnie w zakładce Protection zaznaczyć opcję Permit access with PUT/GET communication...



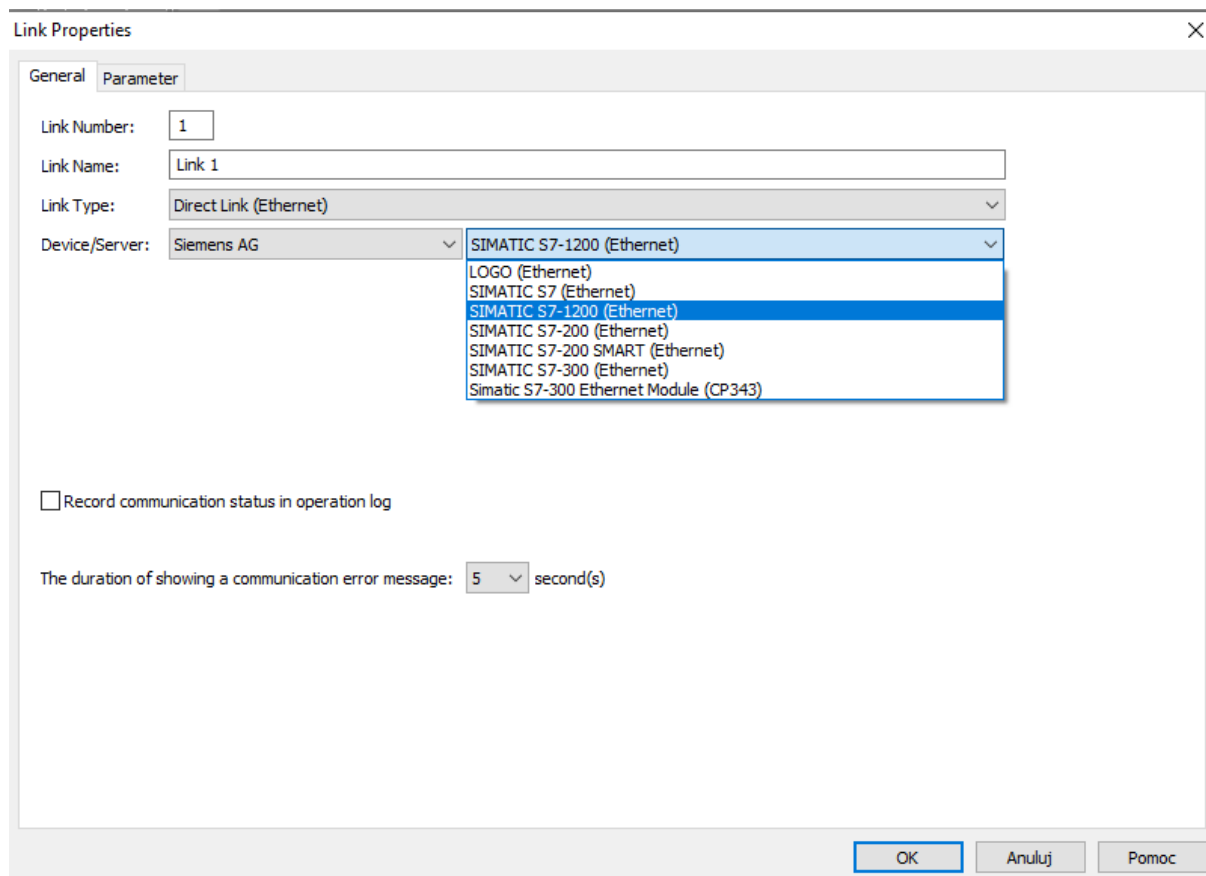
Po wykonaniu tych czynności, skompilowaniu programu i pobraniu aplikacji do sterownika możliwa będzie komunikacja ze sterownikiem z panelu Astraada HMI.

KONFIGURACJA LINKU W PANELU OPERATORSKIM

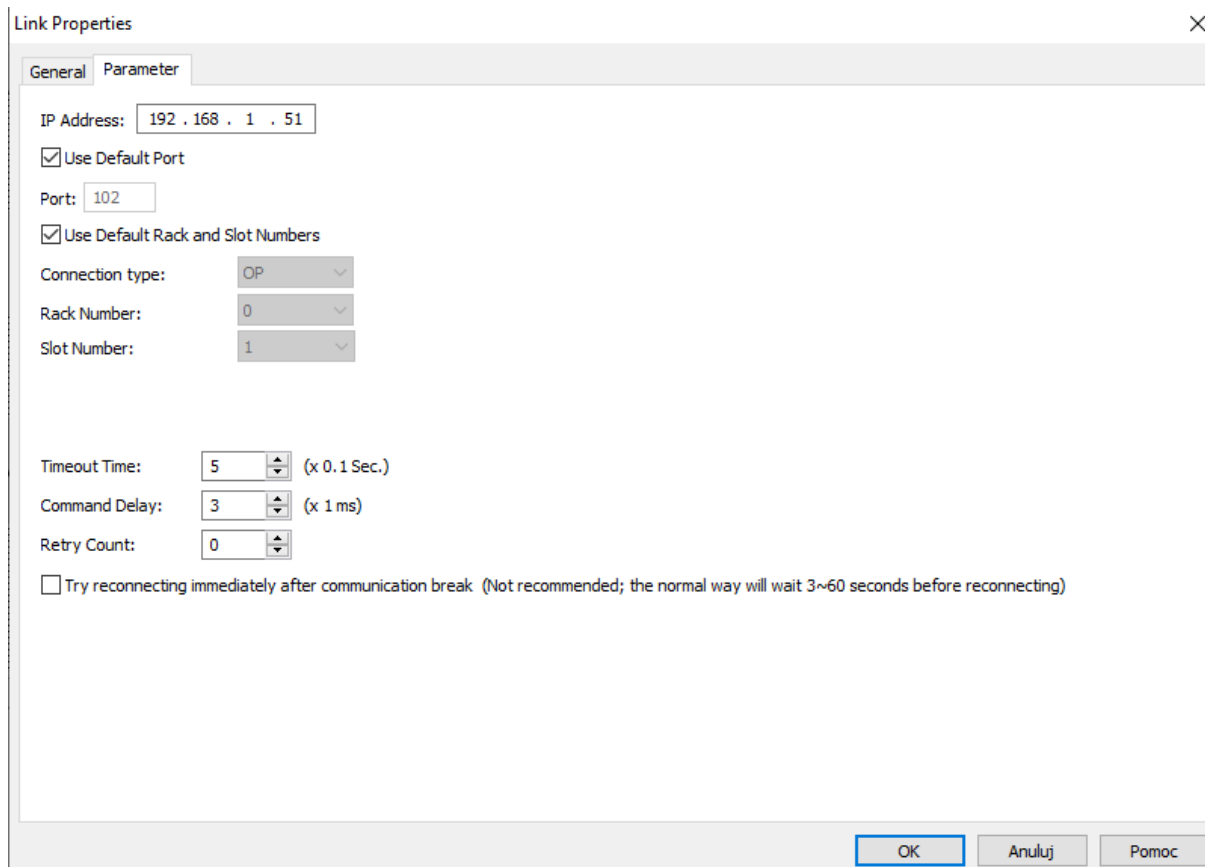
W trakcie tworzenia nowego projektu na panel operatorski w oprogramowaniu Astraada HMI CFG należy skonfigurować link komunikacyjny odpowiedni dla urządzenia S7-1200 w sposób przedstawiony poniżej. Jeśli nie zrobiono tego na etapie tworzenia projektu można dodać nowy link klikając PPM na obiekcie Links w Project Managerze i wybierając Add Link...



W zakładce General należy wybrać link Ethernet, Device/Server Siemens AG -> SIMATIC S7-1200



W zakładce Parameter należy wpisać adres IP sterownika i ustawić niezerowy Timeout Time.



Link Properties

General Parameter

IP Address: 192 . 168 . 1 . 51

Use Default Port

Port: 102

Use Default Rack and Slot Numbers

Connection type: OP

Rack Number: 0

Slot Number: 1

Timeout Time: 5 (x 0.1 Sec.)

Command Delay: 3 (x 1 ms)

Retry Count: 0

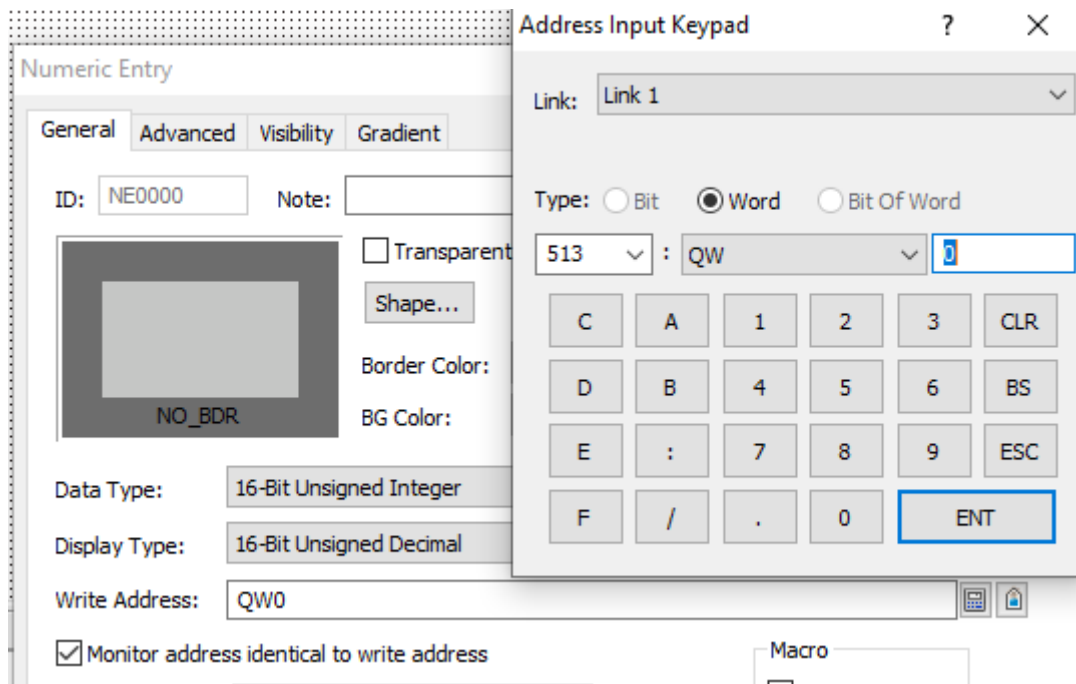
Try reconnecting immediately after communication break (Not recommended; the normal way will wait 3~60 seconds before reconnecting)

OK Anuluj Pomoc

Po zatwierdzeniu ustawień link zostanie skonfigurowany.

ADRESOWANIE ZMIENNYCH

Po podstawowej konfiguracji urządzeń przestawionej powyżej można już odczytywać i zapisywać zmienne w pamięci wewnętrznej sterownika. Dla przykładu utworzono w oprogramowaniu Astraada HMI CFG obiekt Numeric Entry, Bit Button i Bit Lamp. Obiekt Numeric Entry zaadresowano wpisując adres zmiennej QW0 i typ danych 16-Bit Unsigned Integer.



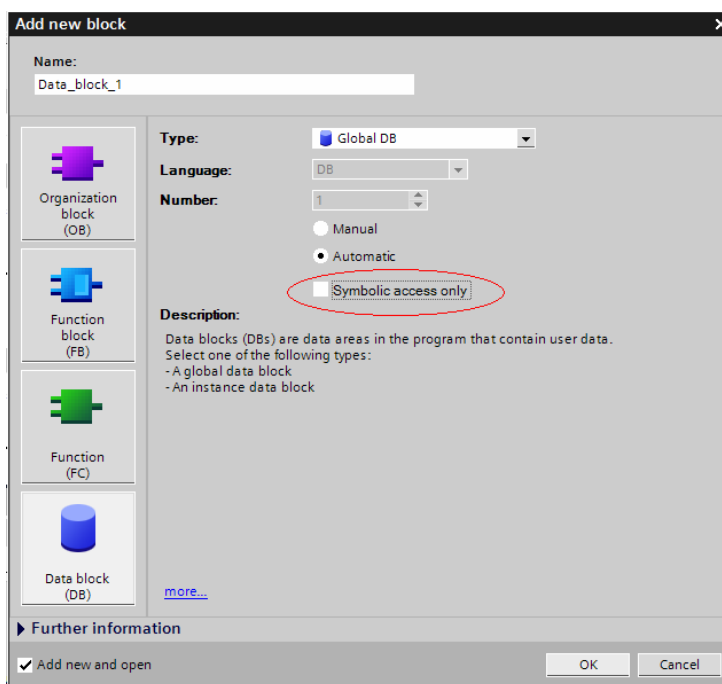
Do obiektu Bit Button w analogiczny sposób przypisano adres Q0.0, a do obiektu Bit Lamp adres I0.0.

Default tag table								
	Name	Data type	Address	Retain	Visibl...	Acces...	Monitor value	Comment
1	qw0	Int	%QW0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	281	
2	i0	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
3	q0	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	TRUE	
4								

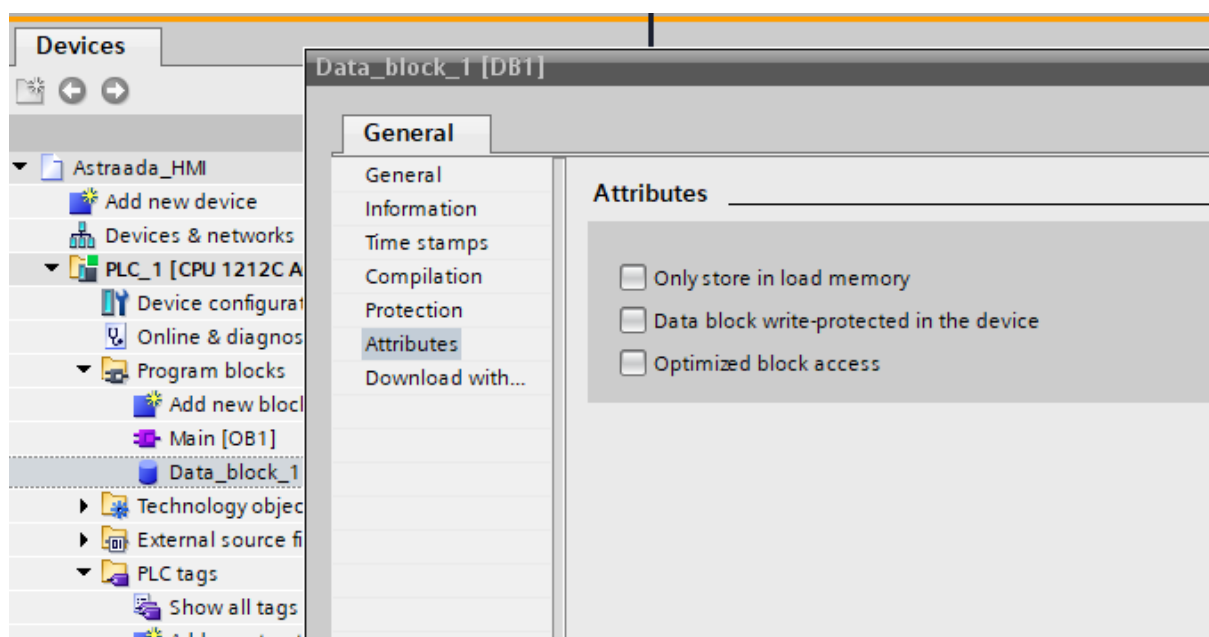
ADRESOWANIE ZMIENNYCH W DATA BLOCKACH

W programie w TIA Portal utworzono dla przykładu nowy Data Block DB1 o nazwie Data_block_1.

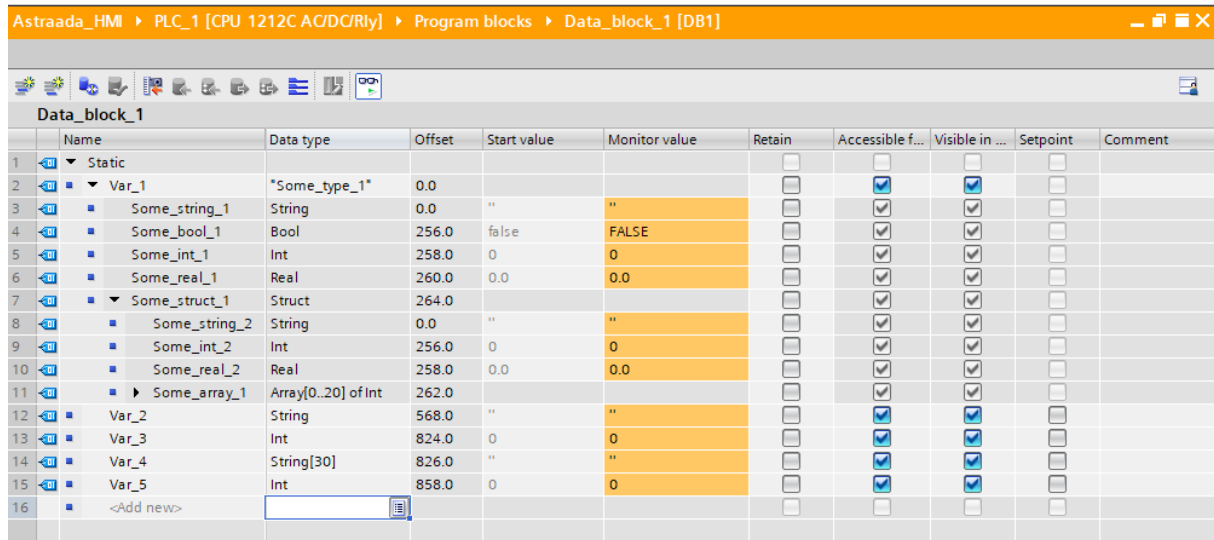
Aby możliwy był odczyt i zapis zmiennych wewnątrz Data Blocków z poziomu panelu operatorskiego podczas tworzenia nowego bloku należy odznaczyć opcję Symbolic access only,



będź w zależności od wersji oprogramowania wejść w ustawienia Data Blocku i w zakładce Attributes odznaczyć opcję Optimized block access.

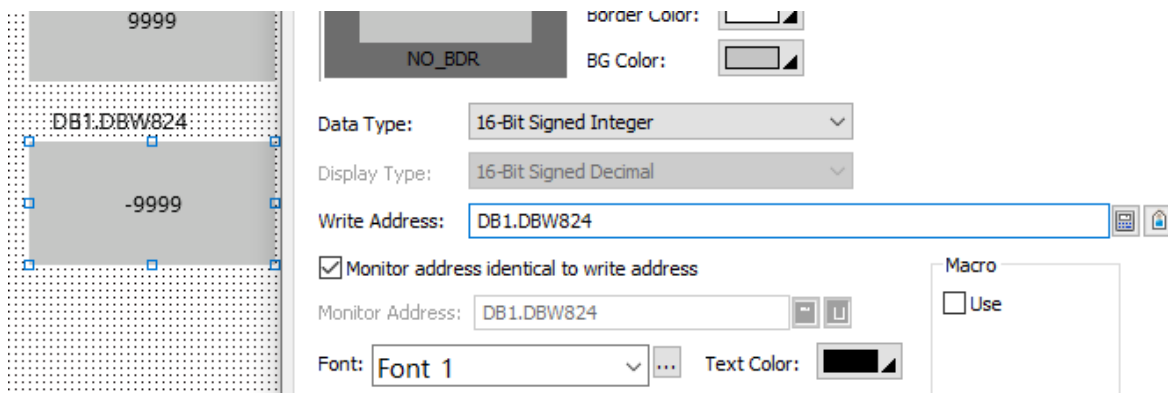


Dzięki temu po kompilacji zmiennym wewnątrz Data Blocków nadane zostaną odpowiednie offsety co później umożliwi ich adresowanie.



	Name	Data type	Offset	Start value	Monitor value	Retain	Accessible f...	Visible in ...	Setpoint	Comment
1	Static									
2	Var_1	"Some_type_1"	0.0				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	Some_string_1	String	0.0		"		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	Some_bool_1	Bool	256.0	false	FALSE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	Some_int_1	Int	258.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	Some_real_1	Real	260.0	0.0	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	Some_struct_1	Struct	264.0				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	Some_string_2	String	0.0		"		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
9	Some_int_2	Int	256.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
10	Some_real_2	Real	258.0	0.0	0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
11	Some_array_1	Array[0..20] of Int	262.0				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
12	Var_2	String	568.0		"		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
13	Var_3	Int	824.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
14	Var_4	String[30]	826.0		"		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
15	Var_5	Int	858.0	0	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
16	<Add new>						<input type="checkbox"/>	<input type="checkbox"/>		

Zmienne wewnątrz data blocków adresować można w oprogramowaniu Astraada HMI CFG „ręcznie”, przykładowo chcąc wymieniać ze sterownikiem wartość zmiennej Var_3 należy użyć składni DB1.DBW824:

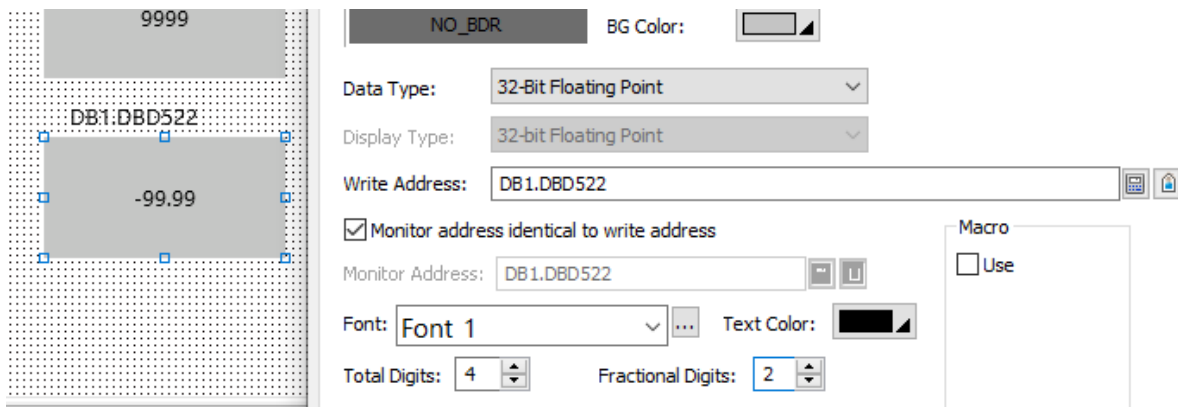


9999
 DB1.DBW824
 -9999

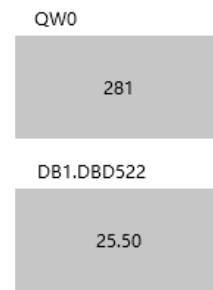
border Color:
 BG Color:
 Data Type: 16-Bit Signed Integer
 Display Type: 16-Bit Signed Decimal
 Write Address: DB1.DBW824
 Monitor address identical to write address
 Monitor Address: DB1.DBW824
 Font: Font 1 Text Color:

Macro Use

W nieco bardziej skomplikowanym przykładzie – chcąc odczytywać zmienną Some_real_2 będącą częścią struktury w ramach własnego typu danych należy obliczyć offset dodając offsety kolejnych poziomów, czyli w tym przypadku offset struktury + offset zmiennej w ramach tej struktury: $264 + 258 = 522$. Dodatkowo w tym przypadku odczytywana zmienna jest zmienną 32-bitową typu floating point, zatem w ustawieniach obiektu należy wybrać odpowiedni typ zmiennej:



1	Static				
2	Var_1	*Some_type_1*	0.0		
3	Some_string_1	String	0.0	"	"
4	Some_bool_1	Bool	256.0	false	FALSE
5	Some_int_1	Int	258.0	0	0
6	Some_real_1	Real	260.0	0.0	0.0
7	Some_struct_1	Struct	264.0		
8	Some_string_2	String	0.0	"	"
9	Some_int_2	Int	256.0	0	0
10	Some_real_2	Real	258.0	0.0	25.5
11	Some_array_1	Array[0..20] of Int	262.0		
12	Var_2	String	568.0	"	"
13	Var_3	Int	824.0	0	55
14	Var_4	String[30]	826.0	"	"
15	Var_5	Int	858.0	0	0



Dodatkowo oprogramowanie Astraada HMI CFG wspiera dynamiczne adresowanie zmiennych w ramach tego drivera w oparciu o Index registers, przykładowo możliwe jest użycie poniższej adresacji:

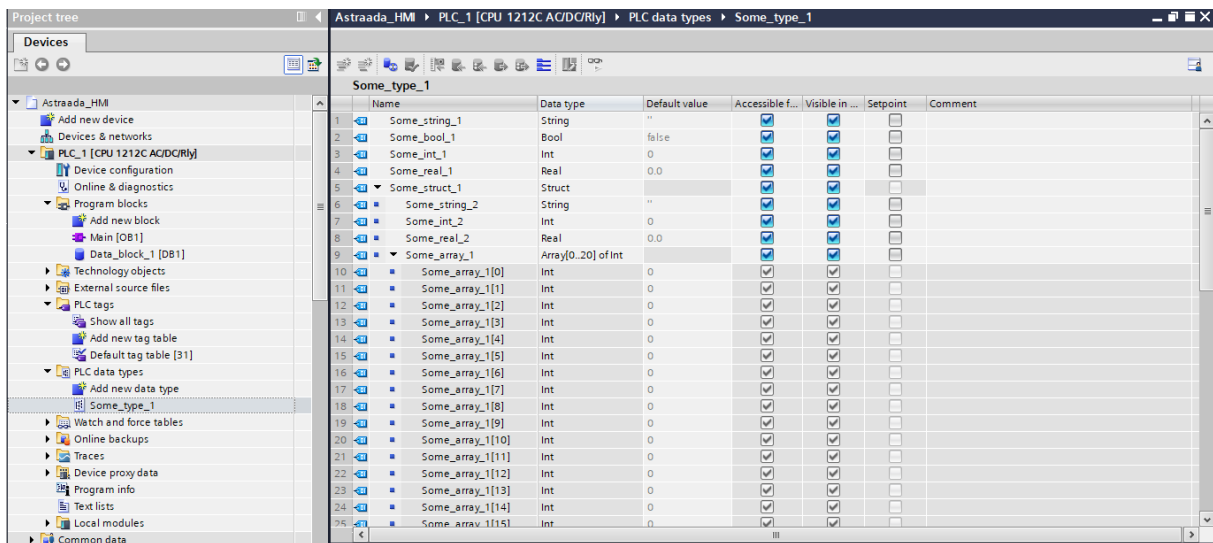
DB[\$I3].DBW100

DB[\$I5+2].DBD[\$I10+16]

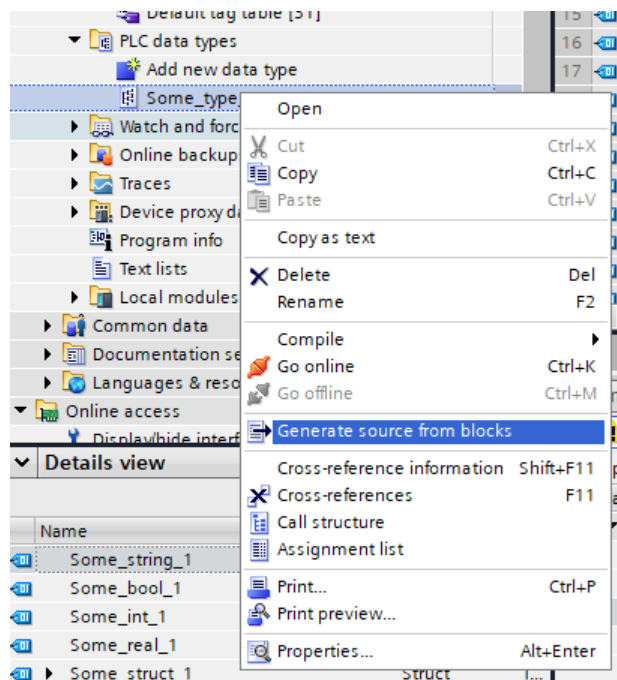
DB[\$I0+2].DBX[\$I1].5

OBSŁUGA WŁASNYCH TYPÓW DANYCH

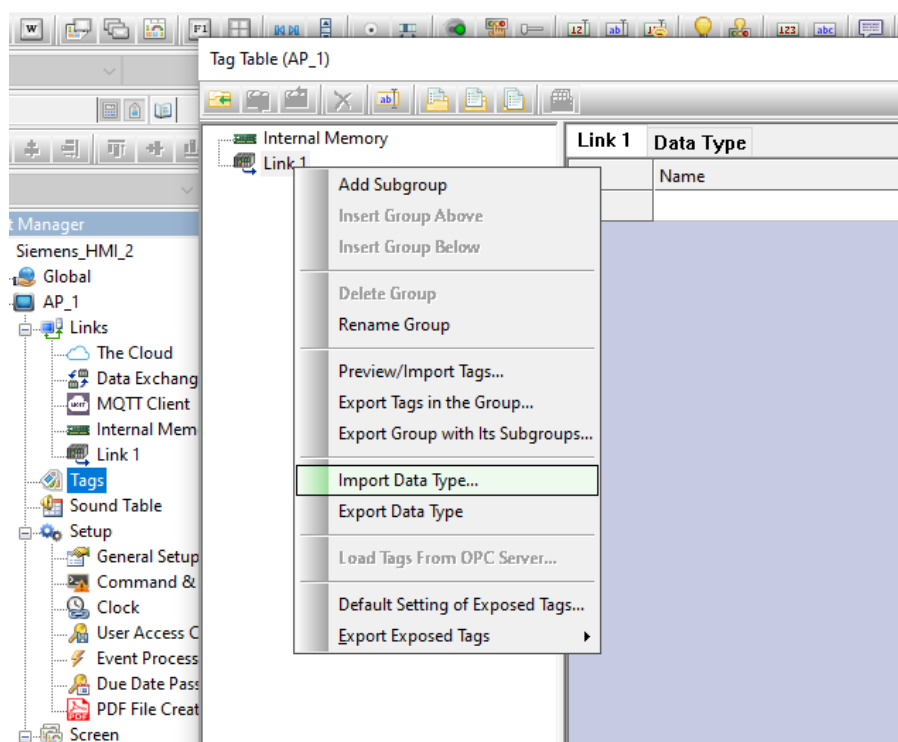
Oprogramowanie Astraada HMI CFG w wersjach 4.0.x.x umożliwia import własnych typów danych tworzonych w oprogramowaniu TIA Portal. Dla przykładu utworzono nowy typ o nazwie Some_type_1.



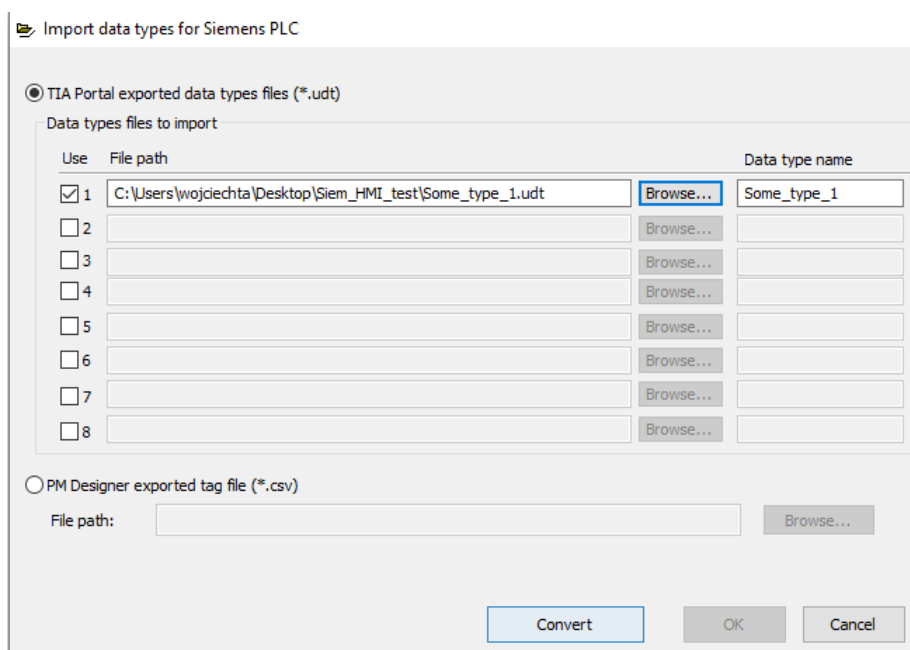
Aby wyeksportować typ danych należy kliknąć PPM na nowym typie i wybrać opcję Generate source from blocks, a następnie zapisać w dowolnym miejscu na dysku komputera plik z rozszerzeniem .udt.



W oprogramowaniu Astraada HMI CFG należy otworzyć okno tagów (klikając dwukrotnie PPM na obiekcie Tags w drzewie projektu) i wybrać z listy po lewej stronie link powiązany ze sterownikiem, kliknąć PPM na tym linku i wybrać opcję Import Data Type...



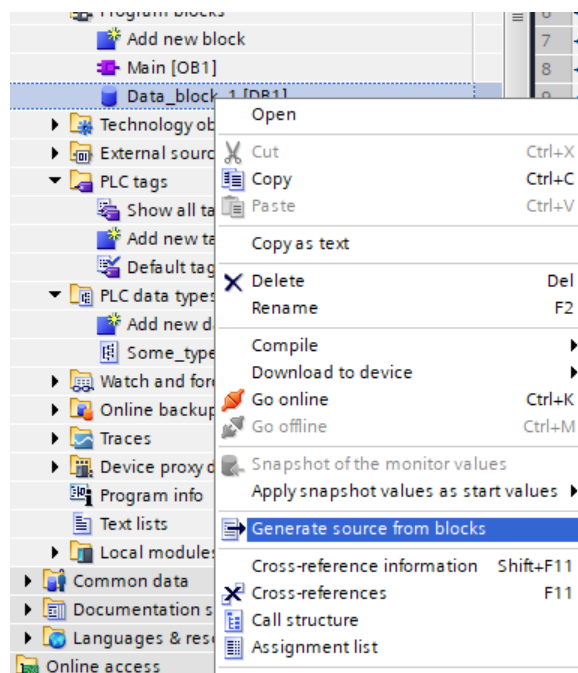
Następnie po zaznaczeniu checkboxa w kolumnie Use należy wybrać odpowiedni plik z dysku i wybrać polecenie Convert.



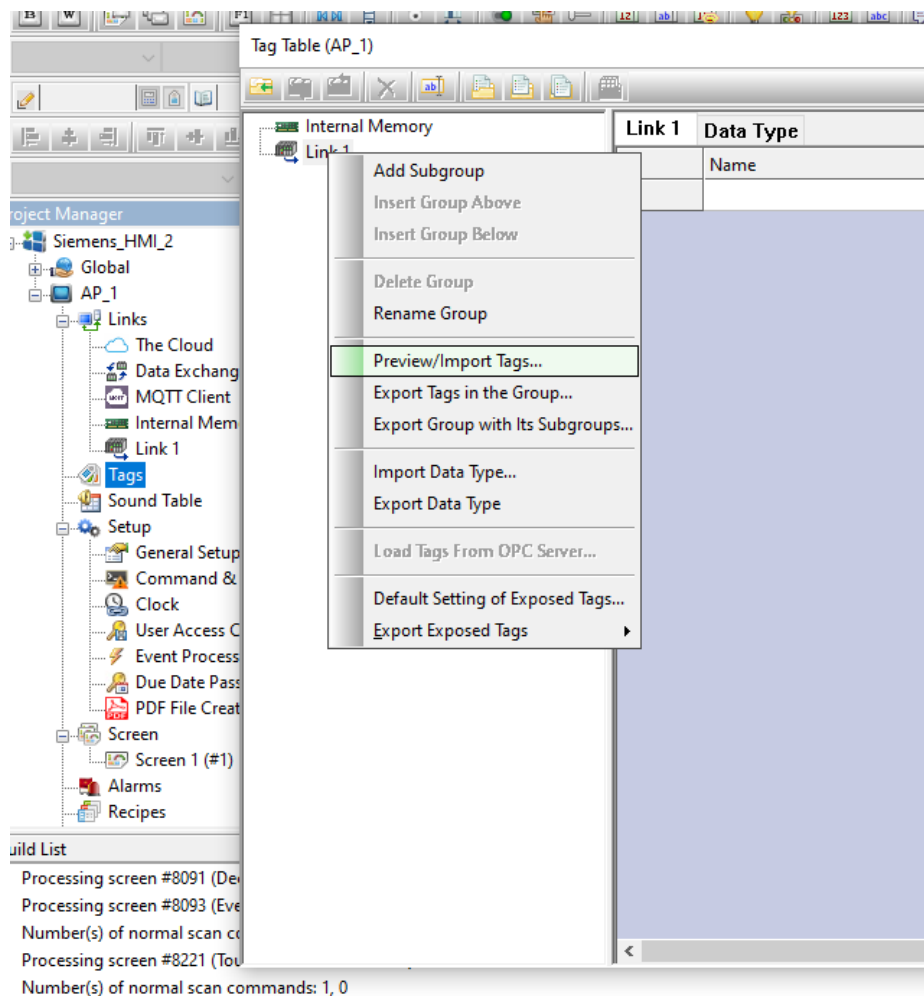
Po tej operacji w części okna po prawej stronie powinna pojawić się kompletna lista zawierająca typ zmiennej, całość należy zatwierdzić przyciskiem OK. Na liście w zakładce Data Type w ramach Tag Table powinny pojawić się zaimportowane typy danych.

IMPORT DATA BLOKÓW

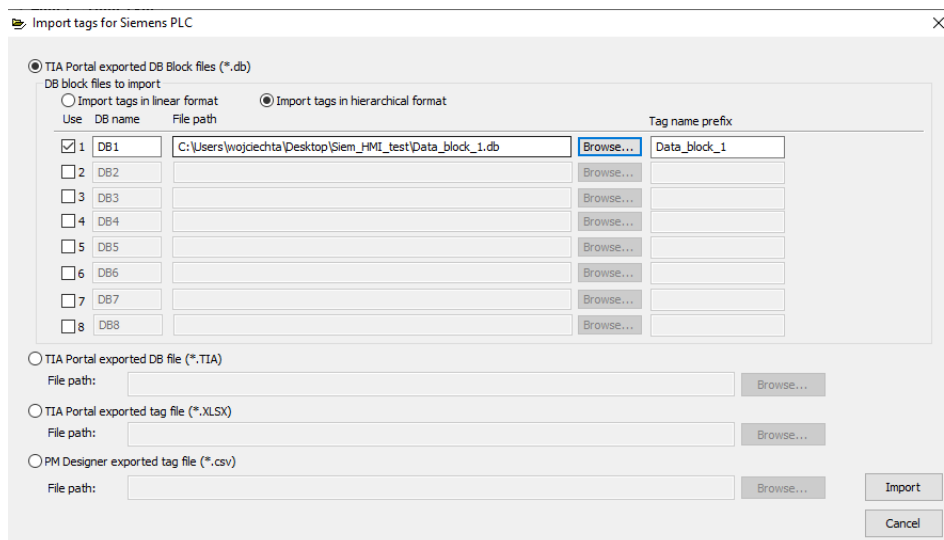
Po utworzeniu Data Blocku zgodnie z wymienionymi wcześniej zaleceniami należy skompilować projekt i wykonać operację Generate source from blocks. Wygenerowany zostanie w ten sposób plik z rozszerzeniem .db, który należy zapisać w dowolnym miejscu na dysku komputera.



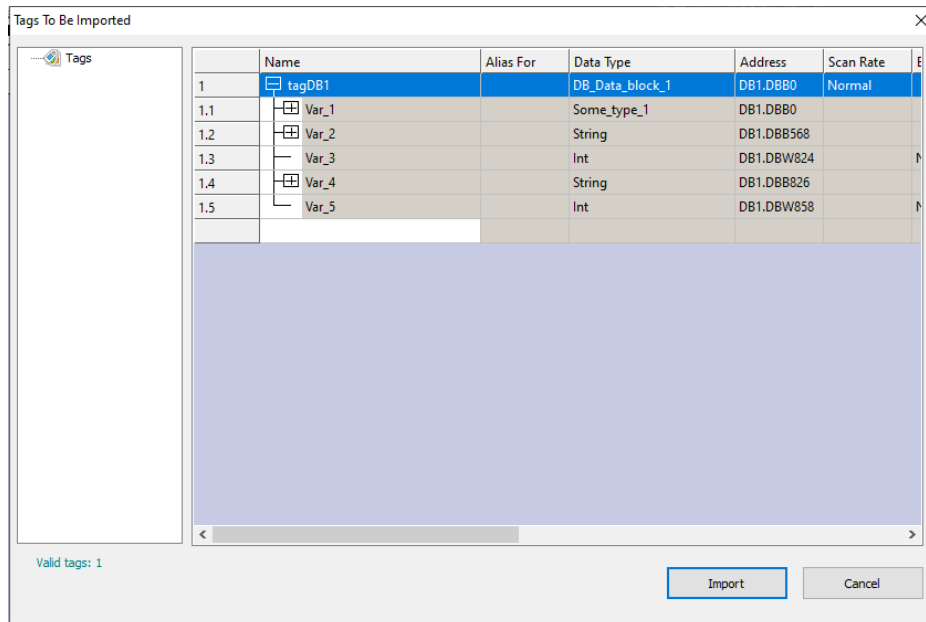
W oprogramowaniu Astraada HMI CFG w oknie Tag Table należy po kliknięciu PPM na odpowiednim linku komunikacyjnym wybrać opcję Preview/Import Tags...



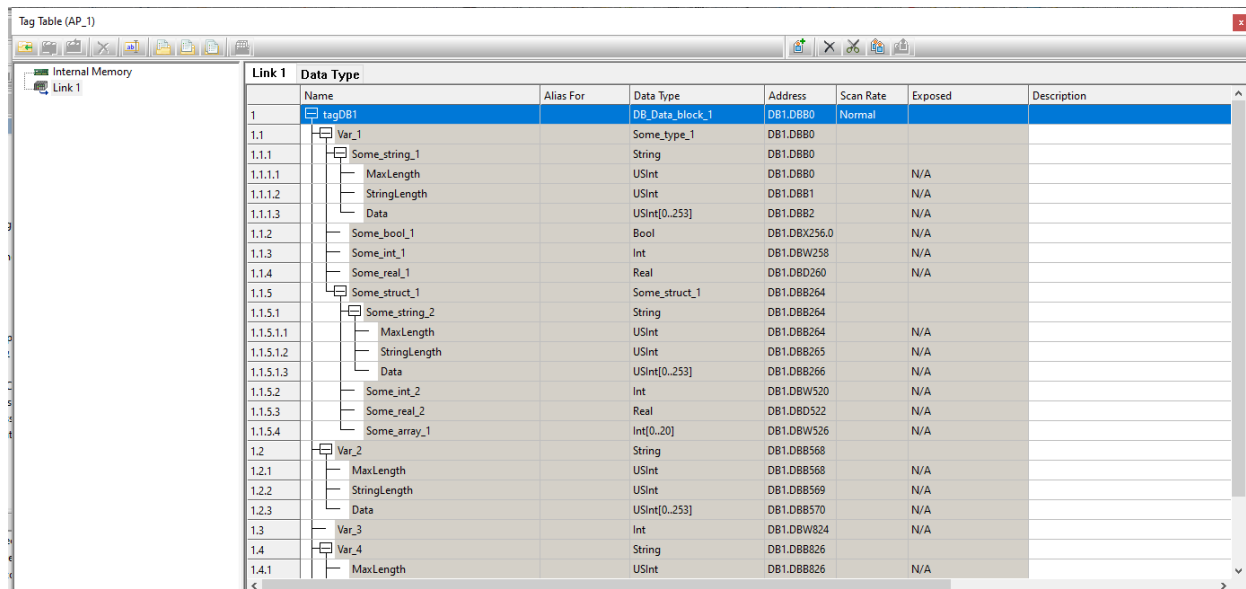
Następnie należy zaznaczyć checkbox w kolumnie Use i wybrać wcześniej wygenerowany plik z rozszerzeniem .db dla Data Blocku z odpowiednim numerem.



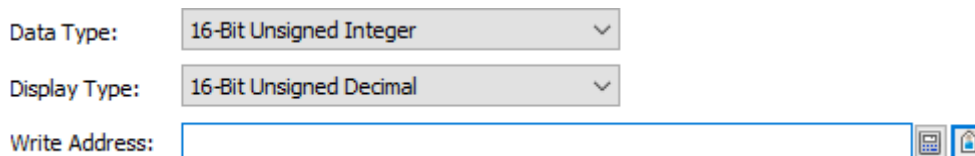
Po zatwierdzeniu przyciskiem Import powinna pojawić się kompletna lista zmiennych, należy kliknąć Import po raz drugi.



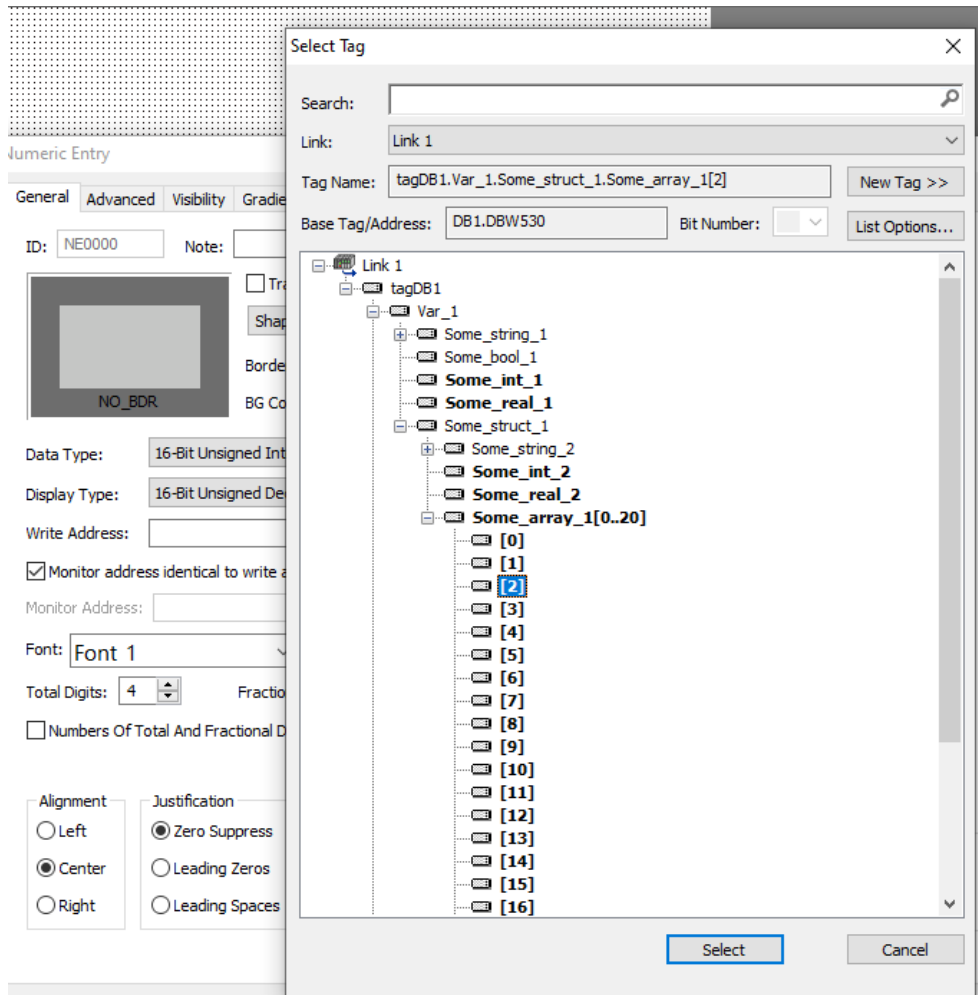
Zmienne zostały dodane do projektu, można już zamknąć okno Tag Table.



Od tego momentu w obiektach w Astraada HMI CFG można odwoływać się do nazw zmiennych, obok pola do wpisywania adresu w ramach obiektów w oprogramowaniu znajduje się przycisk z etykietą:



Po jego użyciu pojawi się okno, w którym będzie można wybrać zmienną z listy, przykładowo chcąc odczytywać wartość z tablicy `Some_array_1` o indeksie 2 w ramach zmiennej `Var_1` wybrano:



Po skompilowaniu i wgraniu programów do urządzeń komunikacja zostanie nawiązana:

7	Some_struct_1	Struct	264.0			
8	Some_string_2	String	0.0	"	"	
9	Some_int_2	Int	256.0	0	0	
10	Some_real_2	Real	258.0	0.0	25.5	
11	Some_array_1	Array[0..20] of Int	262.0			
12	Some_array_1[0]	Int	0.0	0	0	
13	Some_array_1[1]	Int	2.0	0	0	
14	Some_array_1[2]	Int	4.0	0	15	
15	Some_array_1[3]	Int	6.0	0	0	

15