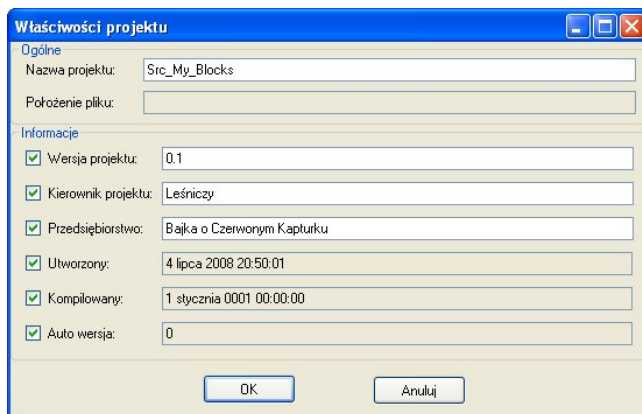
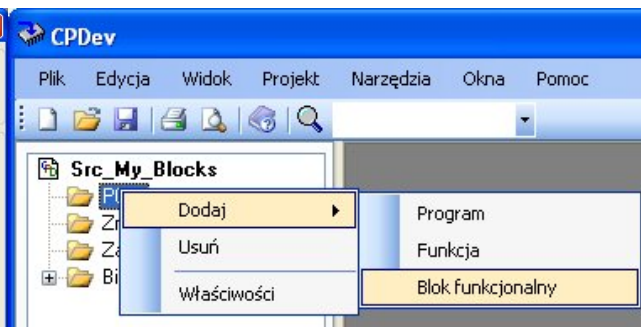


Tworzenie własnych bibliotek w pakiecie CPDev

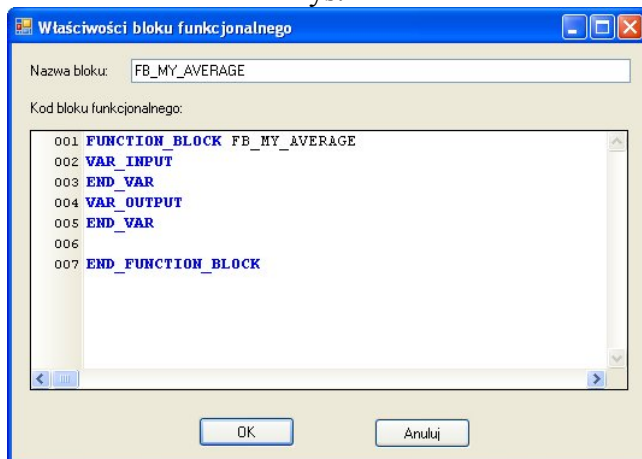
1. Tworzymy nowy projekt (podobnie jak w instrukcji Szybki start).
2. Wypełniamy informacje o projekcie (rys. 1)
3. Dodajemy nowy blok funkcjonalny do projektu (rys. 2)
4. Nadajemy nazwę nowemu obiektowi (rys. 3) i zatwierdzamy przyciskiem OK.
5. Otwieramy edytor i wypełniamy kod bloku funkcjonalnego (rys. 4)
6. Definiujemy nowy blok dla projektu (analogicznie jak na rys. 2)
7. Częściowo wypełniamy kod jak na rys. 5
8. Podglądamy deklarację bloku bibliotecznego – zaznaczamy blok w drzewie RS i naciskamy [Ctrl]+I (rys. 6). Ewentualnie gdy chcemy zobaczyć dokumentację tego bloku to ustawiamy kursor na słowie RS i naciskamy klawisze [Ctrl]+[F1].
9. Uzupełniamy pozostałą część kodu i wyświetlamy odpowiedź do dokończenia nazwy lokalnej zmiennej lub funkcji [Ctrl]+[Spacja] (rys. 7). Warunkiem wyświetlenia odpowiedzi jest kompilacja obecnego projektu.
10. Wypełniamy kod zgodnie z rysunkiem 8.
11. Opcjonalnie możemy przygotować program testowy i wykonać symulację pracy bloków w CPSim
12. Procedura eksportu projektu do biblioteki. Menu 'Projekt | Eksport | Biblioteka' (rys. 9)
13. Wpisujemy nazwę biblioteki My_Blocks oraz opcjonalnie wypełniamy pozostałe pola z prawami autorskimi, położeniem elementów biblioteki w menu (na razie nie obsługiwane) oraz numerem wersji (rys. 10).
14. Klikamy w przycisk wg rys. 11 i wskazujemy katalog oraz nazwę pliku do zapisania pliku bibliotecznego
15. Za pomocą checkboxów znajdujących się obok nazwy obiektów możemy wybrać które składniki z bieżącego projektu zostaną wyeksportowane do biblioteki (rys. 12). Do szybkiego przełączania obiektów (metoda tzw. „szara gwiazdka”) służą przyciski znajdujące się po prawej stronie listy. Przycisk „Wszystkie” przełącza wszystkie typy obiektów; przyciski z ikonami przełączają tylko te typy które odpowiadają ikonom na przycisku (rys. 13). „Interfejs” pozwala na podglądnięcie implementacji danego składnika.
16. Przyciskiem „OK” zatwierdzamy wpisane cechy .
17. Rozpoczyna się kompilacja biblioteki. O jej nieprawidłowym przebiegu świadczą sygnalizowane błędy w polu komunikatów. Nasza biblioteka My_Blocks korzysta z innych bibliotek i dlatego pojawiły się ostrzeżenia podczas jej kompilacji o nie importowalnych zależnościach (rys. 14). Zostanie to usunięte poprzez dodanie zależności dla biblioteki od kompilatorów serii 0.4.11.
18. Zamykamy projekt Src_My_Blocks i tworzymy nowy, który będzie używał naszej biblioteki My_Blocks.
19. Pamiętamy o dodaniu zależnej biblioteki IEC_61131 w przypadku gdy nie została ona dołączona automatycznie.
20. Dodajemy cztery zmienne globalne o nazwach A, B, C, D typu REAL (rys. 15) i analogicznie dwie zmienne typu BOOL – E, F.
21. Tworzymy nowy program i wpisujemy jego kod (rys. 16).
22. Zapisujemy plik i kompilujemy.
23. Uruchamiamy symulator i obserwujemy wyniki pracy programu (rys. 17).



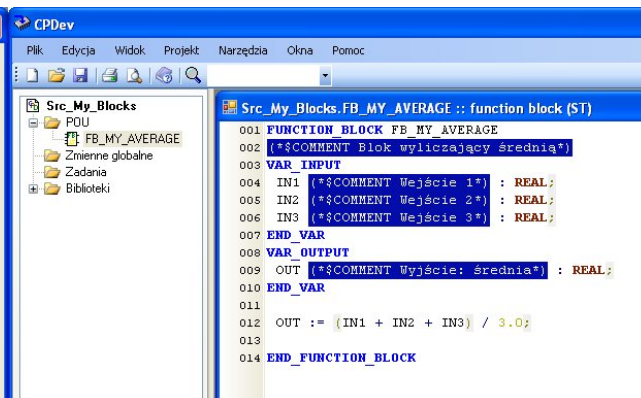
Rys. 1



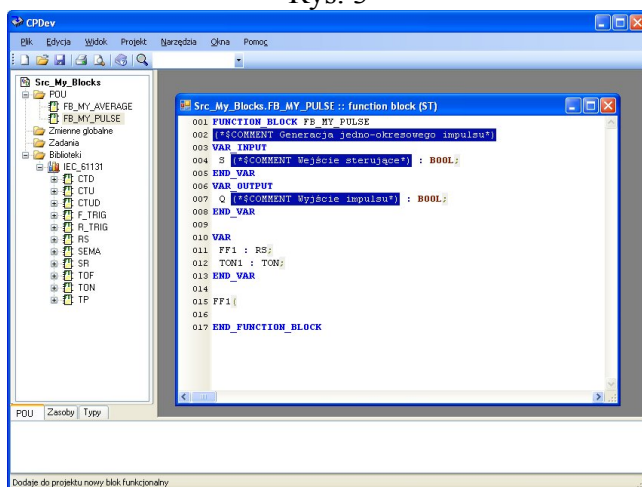
Rys. 2



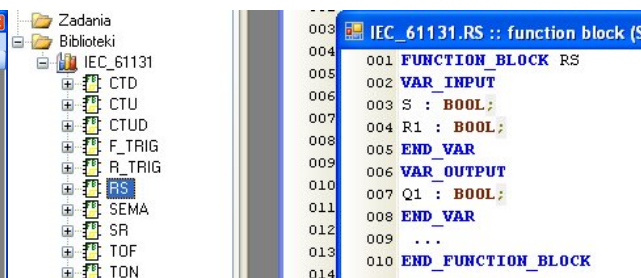
Rys. 3



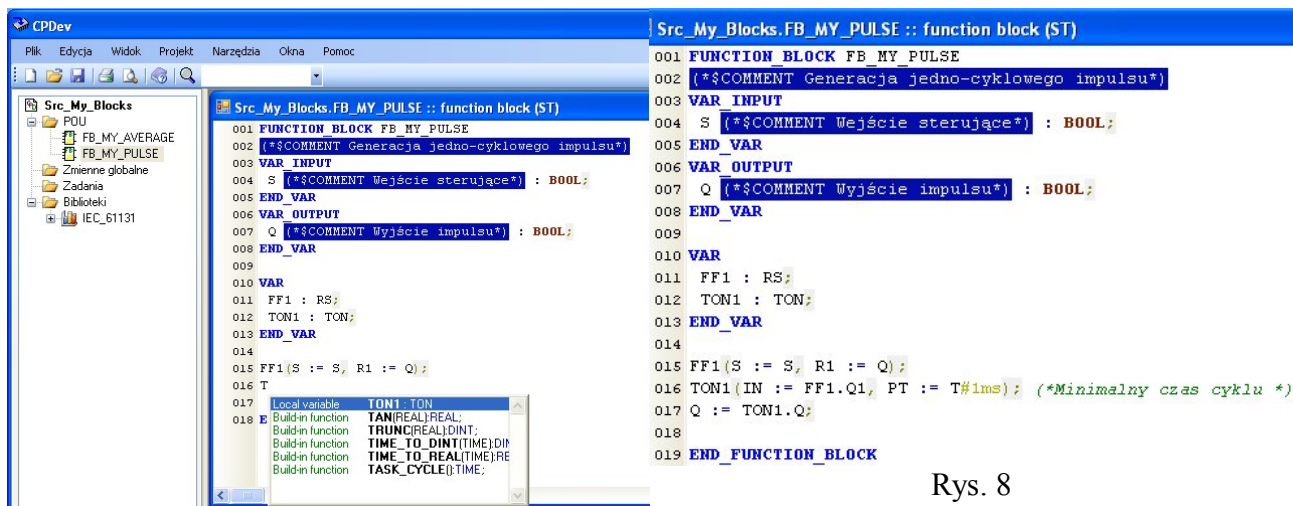
Rys. 4



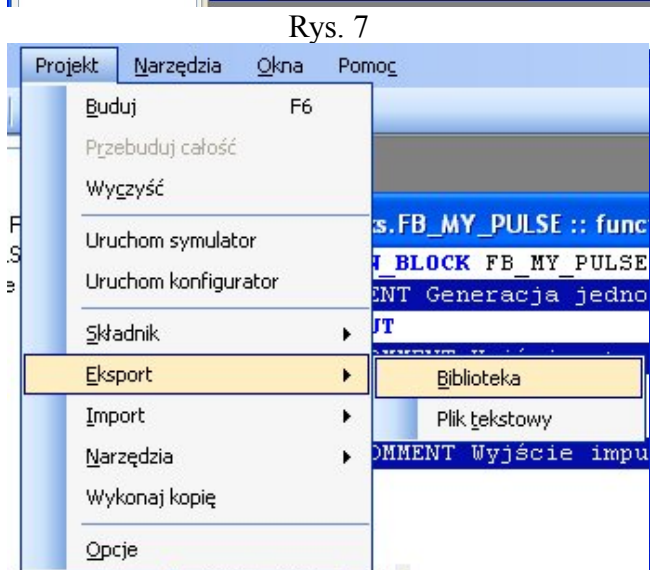
Rys. 5



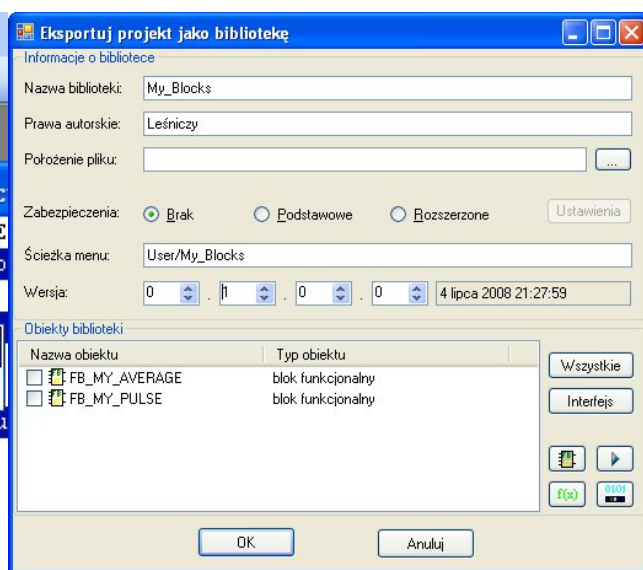
Rys. 6



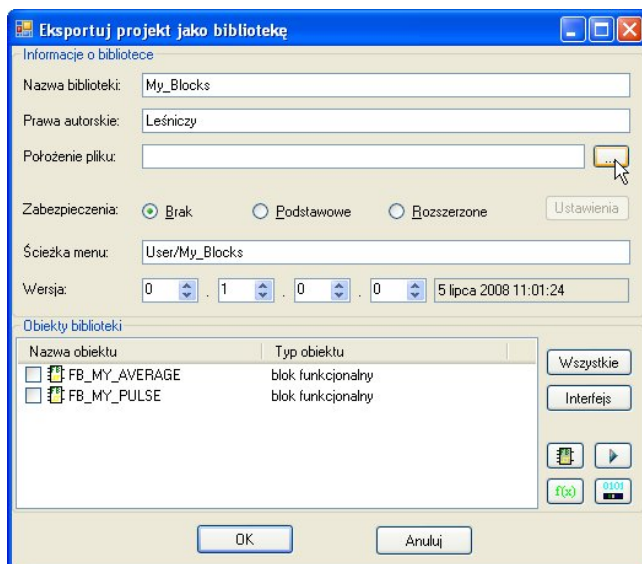
Rys. 8



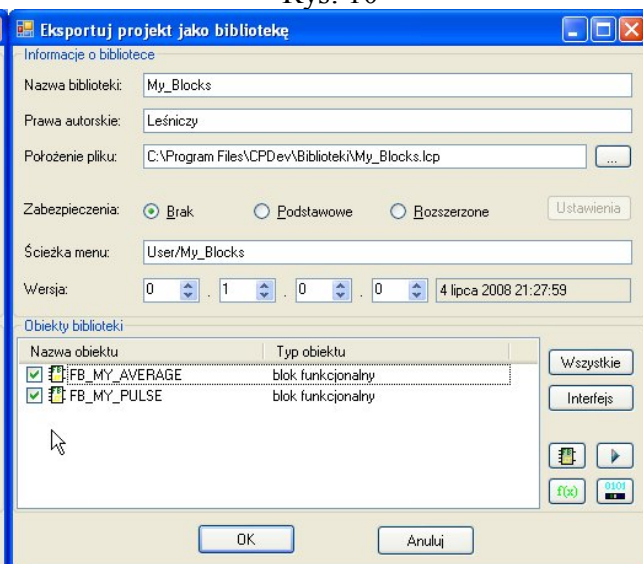
Rys. 9



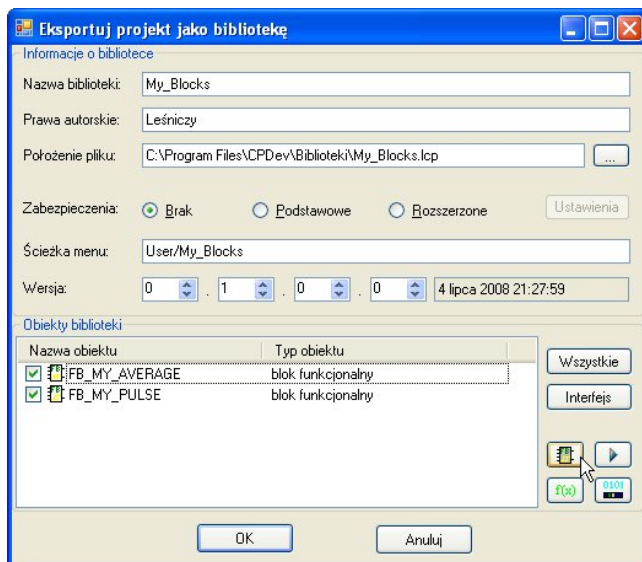
Rys. 10



Rys. 11

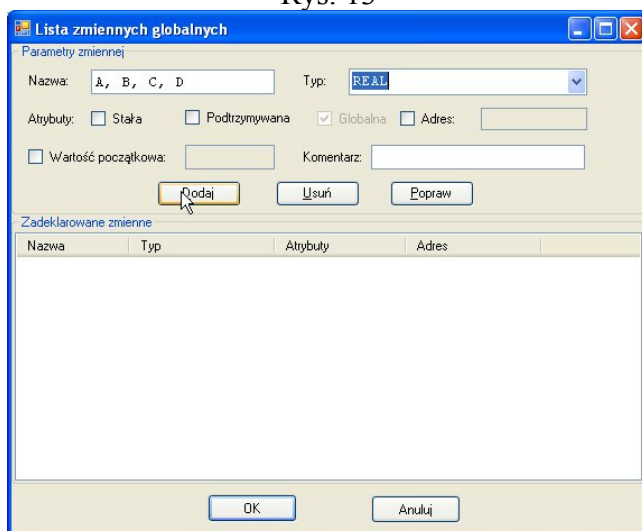


Rys. 12



Rys. 13

Rys. 14



Rys. 15

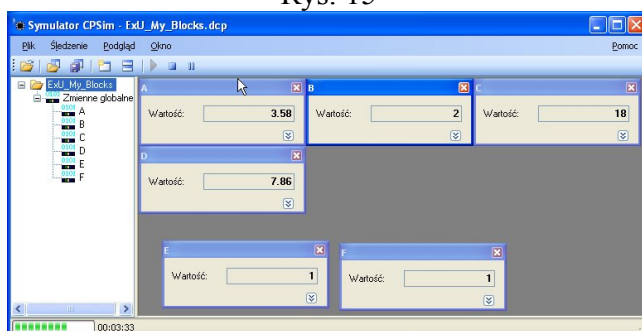
BezNazwy0.PRG_A :: program (ST)

```

001 PROGRAM PRG_A
002 VAR
003   Z : FB_MY_AVERAGE;
004   Y : FB_MY_PULSE;
005 END_VAR
006
007 VAR_EXTERNAL (*$AUTO*)
008 END_VAR
009
010 Z (IN1 := A, IN2 := B, IN3 := C, OUT => D);
011 Y (S := E, Q => F);
012 END_PROGRAM

```

Rys. 16



Rys. 17