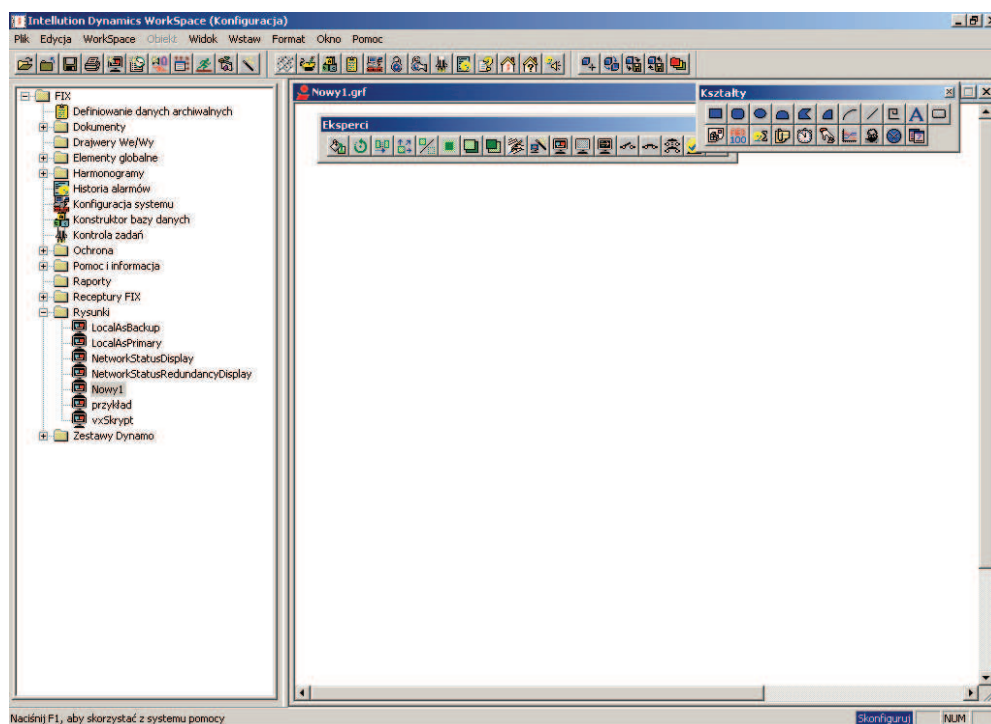


4. Pierwsze kroki

Po uruchomieniu programu iFIX na ekranie pojawi się okno robocze środowiska **WorkSpace** (Rys. 4.1). Na górze okna standardowo znajduje się menu i pasek narzędzi, po lewej stronie umieszczone jest okno zawierające drzewo systemowe, z którego możemy dostać się do wszystkich potrzebnych narzędzi, bibliotek i komponentów, a po prawej stronie umieszczone jest okno edycji w którym buduje się synoptyki wizualizacyjne.



Rys. 4.1

Spróbujmy w szybki sposób stworzyć prostą animację, która wprowadzi nas w środowisko oprogramowania iFIX.

Aby stworzyć dynamiczny element na ekranie musimy wykonać kilka czynności:

- Wprowadzamy do procesowej bazy danych zmienną odzwierciedlającą stan fizycznego urządzenia zewnętrznego. Aby uprościć zadanie wykorzystamy drajwer symulatora **SIM**, który generuje określone sygnały (Dodatek A) i umożliwia budowę i testowanie synoptyk bez konieczności połączenia komputera z rzeczywistymi urządzeniami. W naszym przykładzie wykorzystamy rejestr RH symulatora (sygnał piłokształtny), który będzie zastępował rzeczywisty sygnał z miernika poziomu cieczy w zbiorniku.


System iFIX v3.0 zawiera nowy drajwer symulacyjny, o nazwie SM2, który dostarcza tablicę adresów umożliwiającą testowanie bazy danych procesu pod kątem odpowiedzi bloków i łańcuchów na różne wymuszenia. Wciąż pozostaje dostępny również oryginalny drajwer SIM.

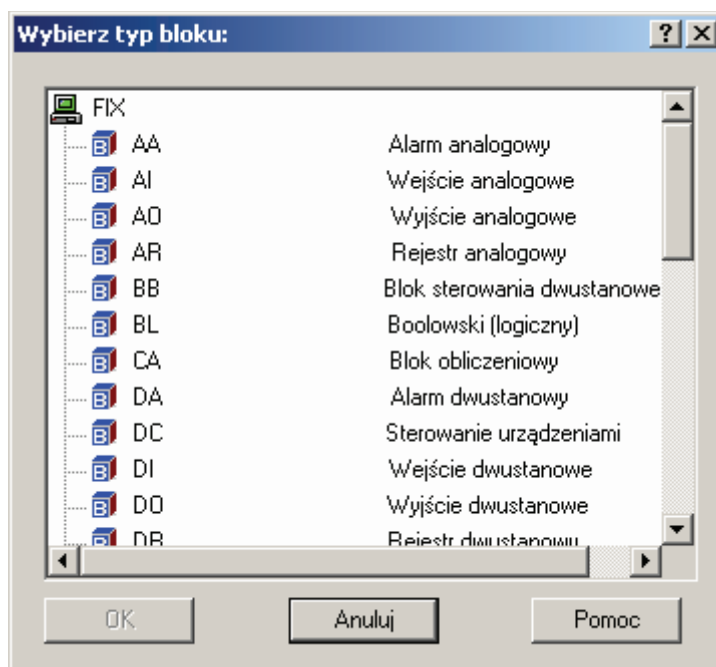
Drajwer SM2 posiada następujące własności:

- Zawiera dwa niezależne zestawy rejestrów. Bloki analogowe uzyskują automatyczny dostęp do rejestrów analogowych, a bloki dwustanowe – do

rejestrów dwustanowych. Zmiana rejestru w jednym zestawie nie powoduje zmiany takiego rejestru w drugim zestawie. Przykładowo, w przypadku zmiany wartości rejestru analogowego o numerze 1000, wartość rejestru dwustanowego o numerze 1000 pozostaje niezmienną.

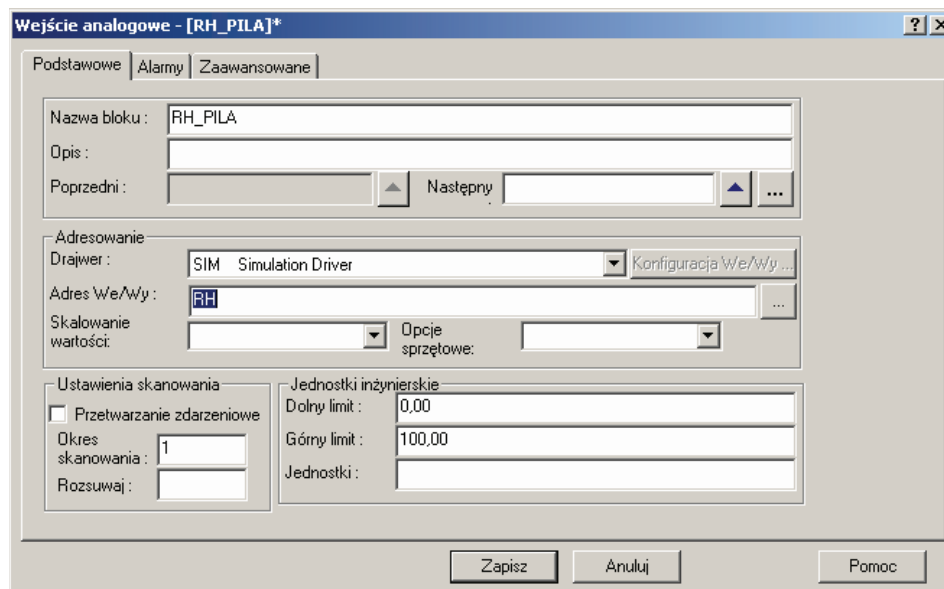
- Zawiera 20000 analogowych oraz 20000 rejestrów 16 bitowych dla rejestrów dwustanowych.
- Zawiera 20000 bajtów danych binarnych lub tekstowych.
- Obsługuje zdarzeniowe przetwarzanie wartości analogowych i dwustanowych.
- Wartości analogowe przechowuje w 4-bajtowych (32-bitowe) rejestrach zmiennoprzecinkowych, numerowanych od 0 do 19999. Wartości wejściowe nie są skalowane. Wartości dwustanowe przechowuje w 16-bitowych rejestrach stałooprzecinkowych, numerowanych od 0 do 19999.
- Zawiera specjalny rejestr do symulacji błędów komunikacji.
- Oferuje interfejs C API umożliwiający aplikacjom dostęp do wartości analogowych i dwustanowych przechowywanych przy pomocy drajwera SM2.
- Obsługuje dane zatrzaśnięte w blokach typu: Wejście analogowe, Alarm analogowy, Wejście dwustanowe, Alarm dwustanowy.
- Może odczytywać i zapisywać status indywidualnych alarmów poszczególnych rejestrów SM2.

- Przyciskiem  z paska narzędzi okna przestrzeni roboczej **WorkSpace** uruchamiamy **Menadżera baz danych (Database Manager)**,
- Potwierdzamy otwarcie **Węzła lokalnego (Local Node)**
- Z menu menadżera bazy danych wybieramy polecenie **Bloki\Dodaj** lub klikamy myszką w pierwszym wolnym polu.
- Określamy rodzaj bloku jako **wejście analogowe AI** (Rys. 4.2)



Rys. 4.2

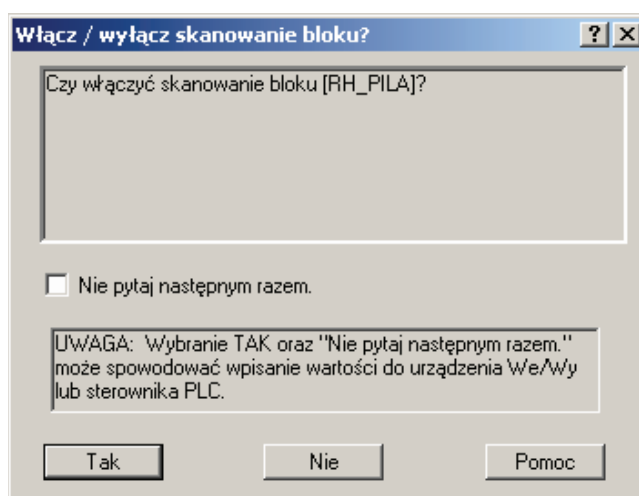
- Wypełniamy okno właściwości danej (Rys. 4.3) - podając w polu **Nazwa Bloku:** nazwę np.: **RH_PILA**, a w polu **Adres We/Wy:** adres **RH**. Pozostałe pola pozostawiamy bez zmian.




Rys. 4.3

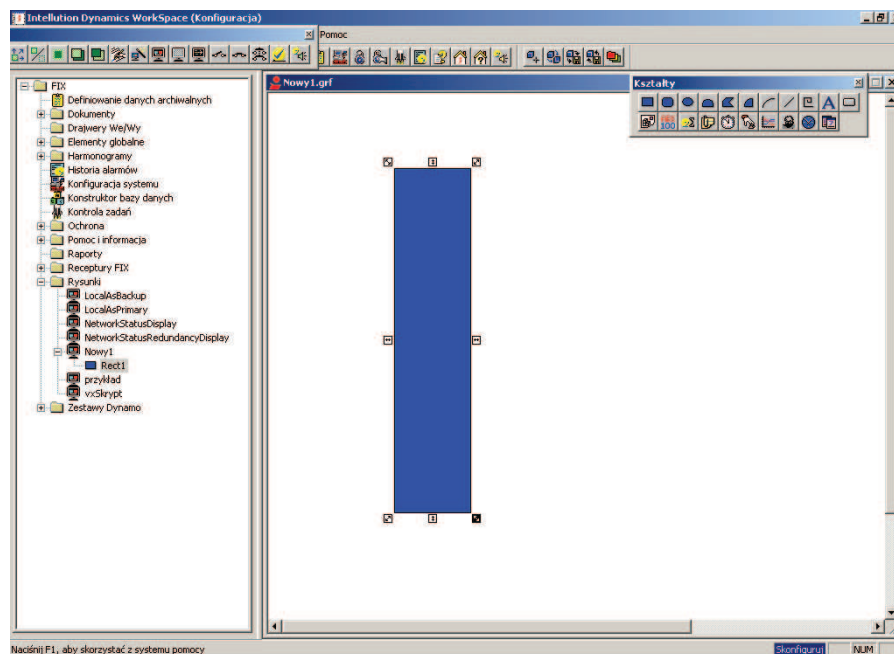
UWAGA: W każdym momencie można skorzystać z pomocy wciskając klawisz F1!

- f) Zapamiętujemy tag przyciskiem **Zapis** (nazwy „tag” = „blok” - będziemy używać dla zmiennych zdefiniowanych w procesowej bazie danych).
- g) Przyciskiem **Tak** (Rys. 4.4) włączamy skanowanie bloku.



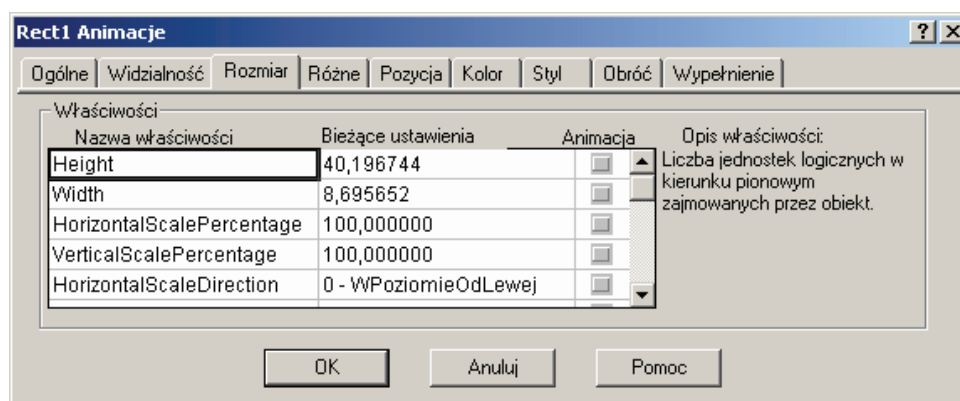
Rys. 4.4

- h) Zapamiętujemy zmiany poleceniem menu **Baza danych/Zapisz** i zamykamy menadżera baz danych.
- Po powrocie do okna przestrzeni roboczej możemy przystąpić do tworzenia animacji
- a) W oknie narzędzi **Kształty** naciskamy symbol prostokąta  (Rectangle) i w oknie edycji rysujemy myszką prostokąt (Rys. 4.5)



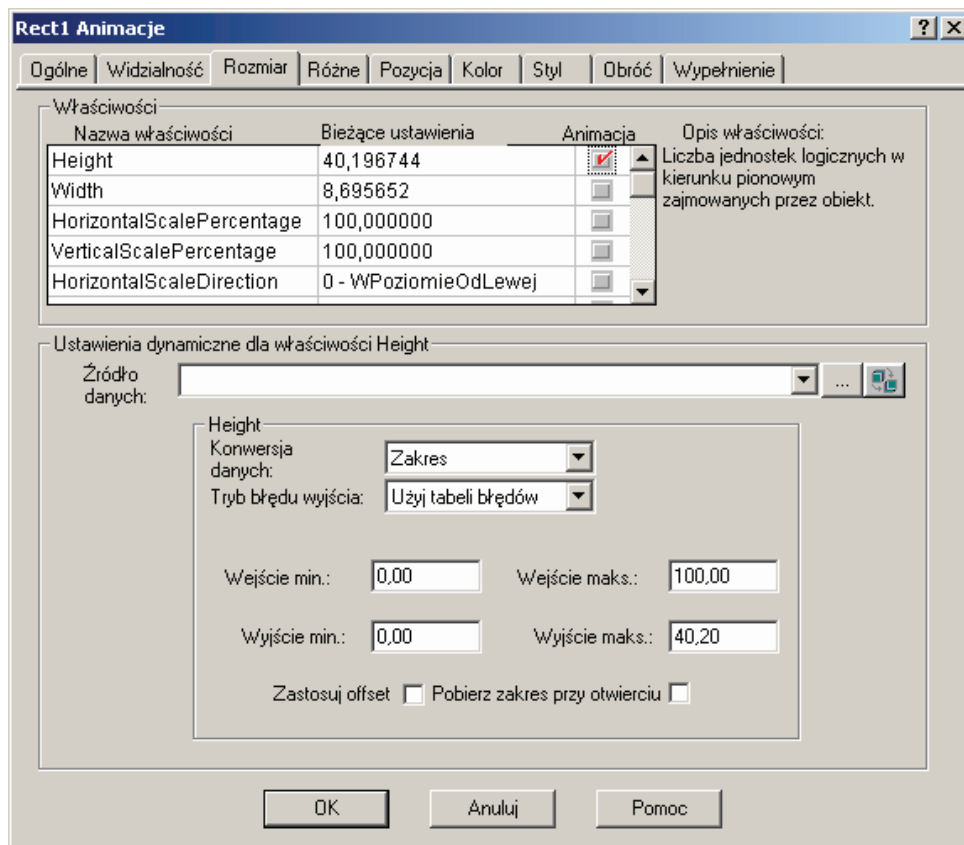
Rys. 4.5

- b) Po dwukrotnym kliknięciu myszką na prostokącie, otworzy się okno właściwości, w którym przechodzimy do zakładki **Rozmiar**, w której możemy sterować wielkością obiektu (Rys. 4.6).




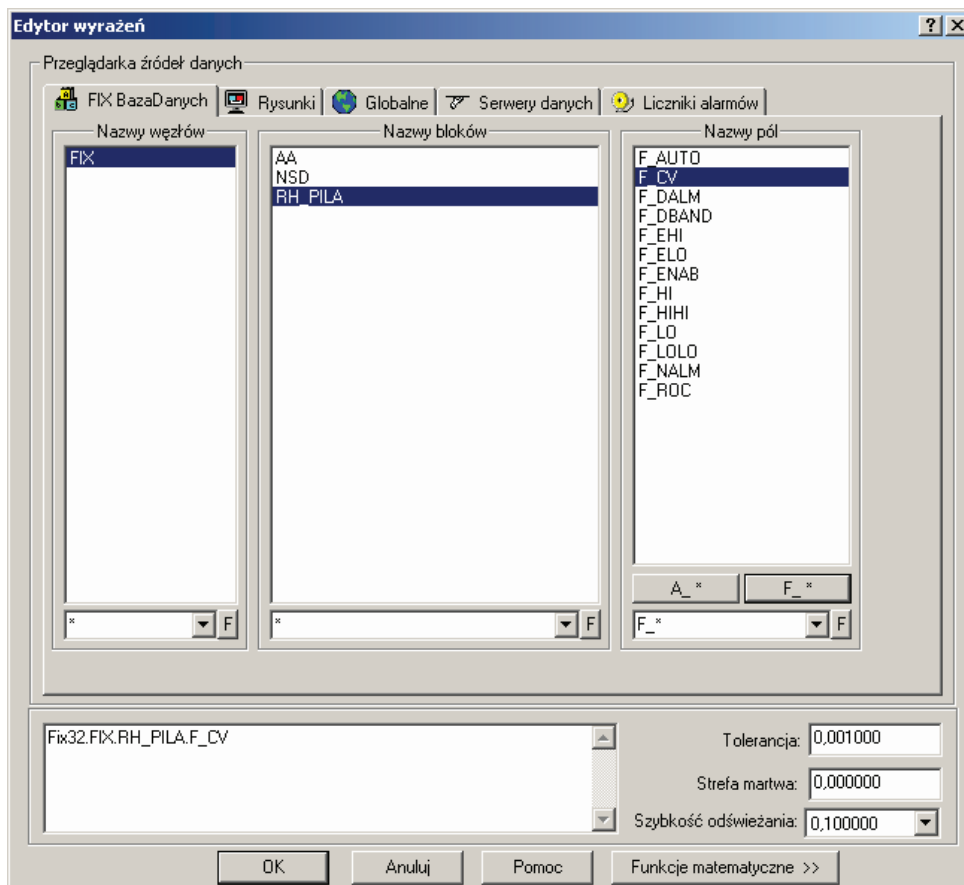
Rys. 4.6

- c) Ponieważ prostokąt ma zmieniać swoją wysokość – zaznaczamy myszką pozycję **Animacja** w pierwszym wierszu **Height**. Okno definicji rozwinie się (Rys. 4.7).



Rys. 4.7

- d) Aby związać wysokość prostokąta z zadeklarowaną wcześniej w bazie zmienną przyciskamy symbol  znajdujący się po prawej stronie pola **Źródło danych**.
- e) W kolejnym oknie (Rys. 4.8) wybieramy zmienną z bazy wybierając:
- w komórce **Nazwy Węzłów** nazwę węzła: **FIX**
 - w komórce **Nazwy Bloków** nazwę tagu: **RH_PILA**
 - w komórce **Nazwy Pól** nazwę pola: **F_CV** (pole to zawiera wartość bieżącą zmiennej (current value) w postaci liczbowej)



Rys. 4.8

f) zamykamy okna potwierdzając zmiany przyciskiem **OK**.

- Po wybraniu z menu **Workspace/Przełącz do trybu wykonywania** (Ctrl+W) nasza animacja zostanie uruchomiona.
- Aby powrócić do trybu projektowego wybieramy z menu **Workspace** polecenie **Przełącz do trybu konfigurowania**.