

ST6-Realizer

część 1



Narysuj swój program!

Za pomocą ST6-Realizera możemy tworzyć proste, jak i bardziej zaawansowane oprogramowanie sterowników ze wspomnianymi mikrokontrolerami. Program ten był już prezentowany na łamach Elektroniki Praktycznej. W związku z niesłabnącym zainteresowaniem Czytelników, postanowiliśmy powrócić do tematu i zaprezentować podstawy projektowania w formie krótkiego kursu.

Każdy elektronik posiadający podstawowe wiadomości z zakresu techniki cyfrowej nie powinien mieć większych trudności z „ujarzmieniem“ programu. Przygotowywanie programu dla mikrokontrolera to kilka przyjemnych chwil spędzonych przy komputerze. Praktycznie najprostszy program możemy stworzyć w kilka minut.

Zaczynamy oczywiście od zainstalowania programu, którego jedna z wer-



Opracowany przez holenderską firmę Actum system graficznego tworzenia programów pod nazwą Realizer zdobył spore uznanie użytkowników na całym świecie. Od chwili opublikowania przez nas ST6-Realizera na płycie CD-EP2 pojawiło się już kilka nowych wersji tego programu, także dla innych mikrokontrolerów niż ST62. Ponieważ program nie jest dostępny w polskiej wersji, przygotowaliśmy dla Was krótki kurs obsługi pakietu, którego „rdzeniem“ będzie przykładowy projekt. Zaczynamy od krótkiego wstępu, którego zadaniem jest przybliżenie zasad posługiwania się programem.



Rys. 1.

sji znajduje się na płycie CD-EP2 (płyta w ofercie AVT). Publikujemy ją także na CD-EP2/2001B w katalogu \Kurs. Minimalne wymagania sprzętowe programu nie są wygórowane:

- ✓ system operacyjny WIN3.11/9x/2000/Me,
- ✓ procesor 486 lub lepszy,
- ✓ pamięć operacyjna min. 8MB,
- ✓ ok. 14MB wolnego miejsca na dysku twardym.

Instalację programu rozpoczynamy od uruchomienia programu instalacyjnego, który znajduje się w katalogu \Kurs (płyta CD-EP2/2001B). Podczas instalacji zostanie utworzony folder C:\Realizer, w którym umieszczane są wszystkie rozpakowywane pliki programu wraz z przykładami. Aby zain-



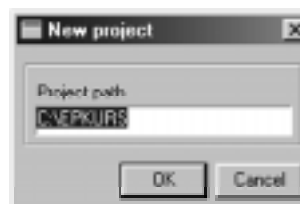
Rys. 2.

stalowany program poprawnie pracował, należy od razu po instalacji zrestartować komputer.

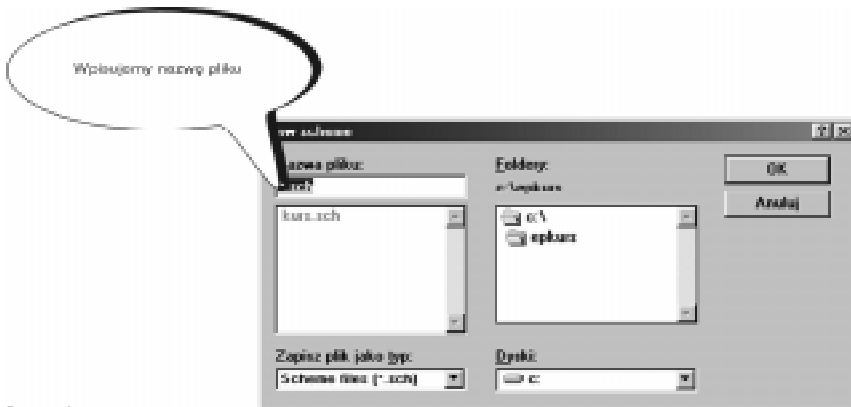
Pakiet programów wchodzących w skład Realizera składa się z czterech części, niezbędnych do tworzenia projektów. Są to:

- ✗ Realizer,
- ✗ Simulator,
- ✗ Analityzer,
- ✗ Sch2Lib Converter.

Realizer jest głównym programem pakietu, za pomocą którego możemy przeprowadzić edycję tworzonego oprogramowania i otworzyć wszystkie programy pomocnicze. Do przeprowadzenia analizy działania tworzonego



Rys. 3.

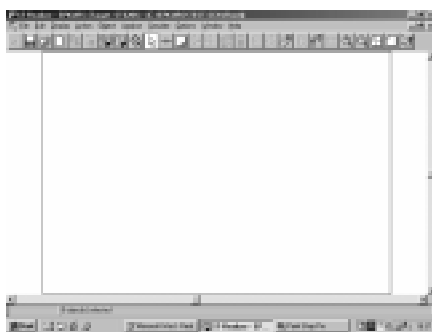


Rys. 4.

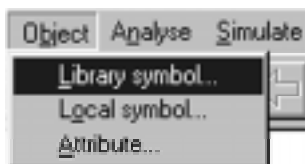
oprogramowania jest przeznaczony program *Analyser*, który generuje plik programu wynikowego w postaci szesnastkowej oraz plik raportu z wykazem ewentualnych błędów. *Simulator* umożliwia symulację oprogramowania sterownika po jego analizie. *Sch2Lib-Converter* jest przeznaczony do konwersji schematów do postaci modułów bibliotecznych, co umożliwia rozszerzenie zawartości biblioteki.

Standardowo Realizer oferuje nam bogaty zestaw elementów bibliotecznych. Są w niej zawarte wszystkie podstawowe elementy logiczne, takie jak AND, OR, INV, EXOR, NOR i NAND o różnej liczbie wejść. Oprócz elementów logicznych biblioteka zawiera szereg modułów funkcjonalnych m.in.:

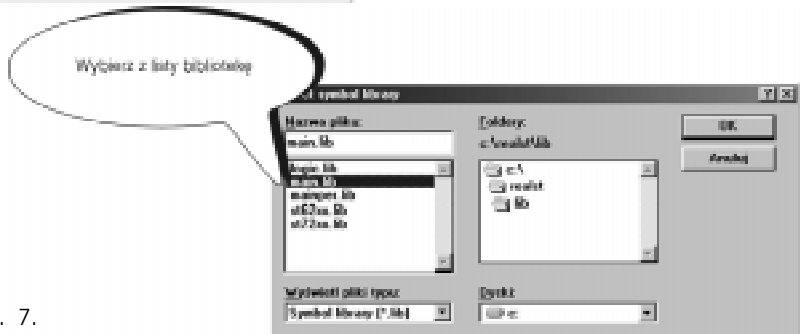
- komparator,
- generator opóźnienia,
- wyjście cyfrowe,
- wejście cyfrowe i analogowe,
- liczniki dwukierunkowe,
- multipleksery,
- detektory zmian stanów logicznych,
- detektory pojawienia się zbocza,
- przetwornik analogowo-cyfrowy,
- stałe,



Rys. 5.



Rys. 6.



Rys. 7.

- układy arytmetyczne: sumator, subtraktor, moduł dzielący i mnożący,
- przerzutnik D.

Jak tworzy się projekt?

Cały proces możemy podzielić na następujących pięć etapów (rys. 1):

- pomysł, czyli sformułowanie zadania,
- narysowanie schematu oprogramowania,
- kompilacja,
- testowanie oprogramowania,
- zaprogramowanie mikrokontrolera.

Zanim jednak przystąpimy do tworzenia oprogramowania sterownika, zapoznajmy się z programem. Po jego uruchomieniu otwiera się główne okno programu (rys. 2). W menu *File* wybieramy *New Project* i otwiera się okno dialogowe pozwalające ustalić ścieżkę dostępu do plików projektu (rys. 3). W *Project path* wpisujemy np. *c:\EPKURS* i następnie klikamy OK. Na dysku *C:* zostanie utworzony katalog o nazwie *EPKURS*, w którym będą zapisywane wszystkie pliki związane z naszym projektem.

Kolejnym krokiem jest utworzenie pliku schematu. W menu *File* klikamy *NEW*, otwiera się okno *New Scheme* (rys. 4), w którym wpisujemy nazwę pliku (np. *KURS*) i naciskamy OK. Po tej operacji zostanie utworzony w folderze *EPKURS* plik *KURS* i otworzy się czysta strona do edycji schematu programu (rys. 5).

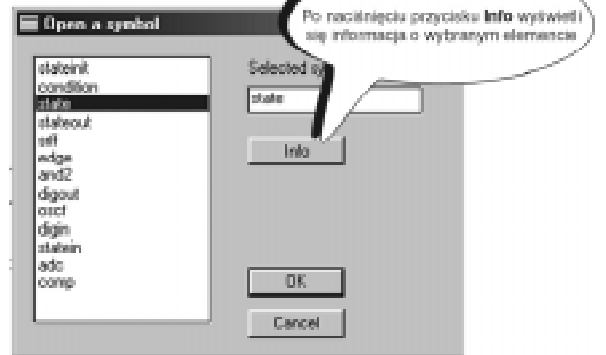
Od tego momentu można rozpocząć realizację projektu, której przebieg szczegółowo omówimy za miesiąc.

Pasek narzędziowy

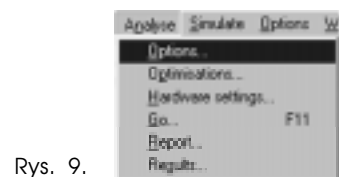
W pasku górnego menu najbardziej interesujące są: *Object*, *Analyse*, *Options*.

W menu *Object* mamy trzy polecenia (rys. 6):

- *Library symbol* - dokonujemy wyboru biblioteki, z której będziemy korzystać przy tworzeniu programu.



Rys. 8.



Rys. 9.



Rys. 10.



Rys. 11.



Rys. 12.



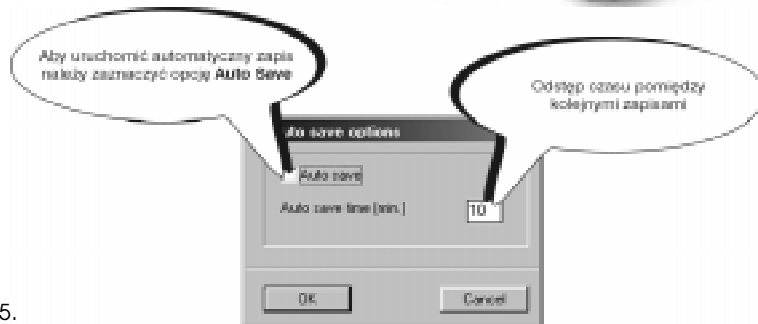
Rys. 13.

Na początku proponuję wybrać bibliotekę *main.lib* zawierającą wszystkie elementy użyte w projekcie oprogramowania (rys. 7).

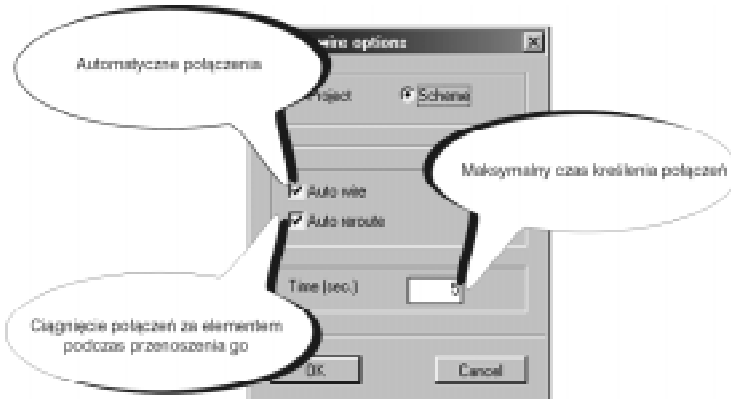
- *Local symbol* - automatycznie tworzy własną bibliotekę z elementów co najmniej raz użytych w danym projekcie. Jest to funkcja ułatwiająca pracę z programem. Po naciśnięciu OK wybrany element zostaje umieszczony na planszy schematu programu (rys. 8).
- *Attribute* jest to polecenie programu do nadawania charakterystycz-



Rys. 14.



Rys. 15.



Rys. 16.

nych właściwości symbolom i połączeniom między symbolami. Funkcja ta jest wykorzystywana przy tworzeniu bardziej zaawansowanych programów.

W menu *Analyse* mamy następujące polecenia (rys. 9):

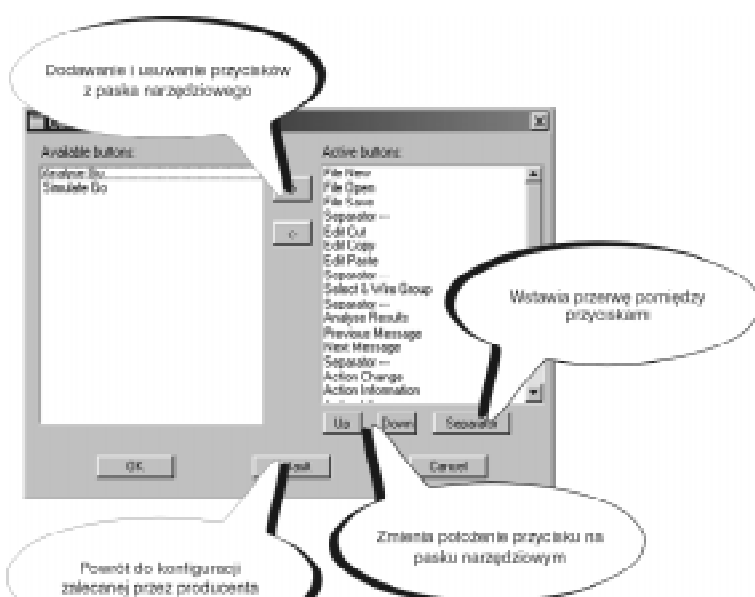
- *Options* - ustawiamy parametry do przeprowadzenia analizy tworzonego oprogramowania (rys. 10).

Za pomocą *Hardware settings* (rys. 11) możemy dokładnie ustawić parametry do analizy dla wybranego przez projektanta typu mikrokontrolera. Zaleca się jednak, aby nie zmieniać ustawień domyślnych na początku pracy z Realize-

rem. Polecenie *Go* uruchamia program z Pakietu Realizera - ST-Analyzer - tworzący plik wynikowy w kodzie szesnastkowym (rys. 12). Po naciśnięciu polecenia *Report* zostanie w notatniku wyświetlony plik raportu, w którym znajdują się wszystkie informacje dotyczące przebiegu kompilacji. Uruchomienie polecenia *Results* wyświetli ewentualne błędy, jakie powstały podczas analizy oprogramowania (rys. 13).

W menu *Options* (rys. 14) umieszczono następujące polecenia:

- w *Auto save* ustawiamy czas automatycznego zapisu wszelkich zmian w schemacie (rys. 15). Program do-

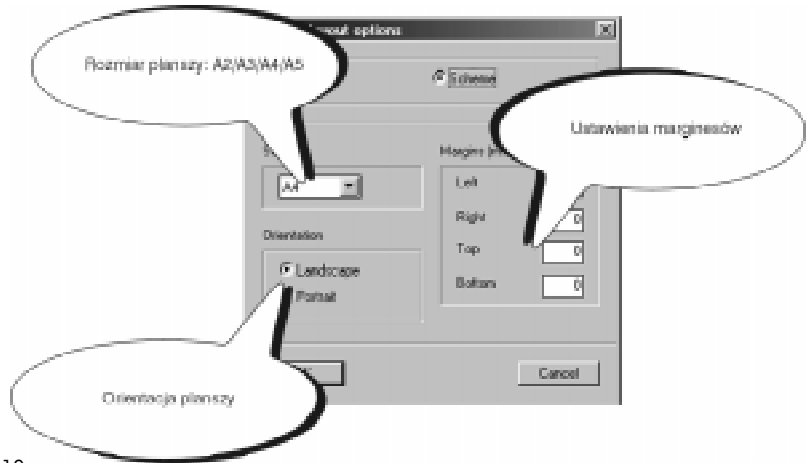


Rys. 17.

myślnie ma ustawiony zapis co 10 minut.

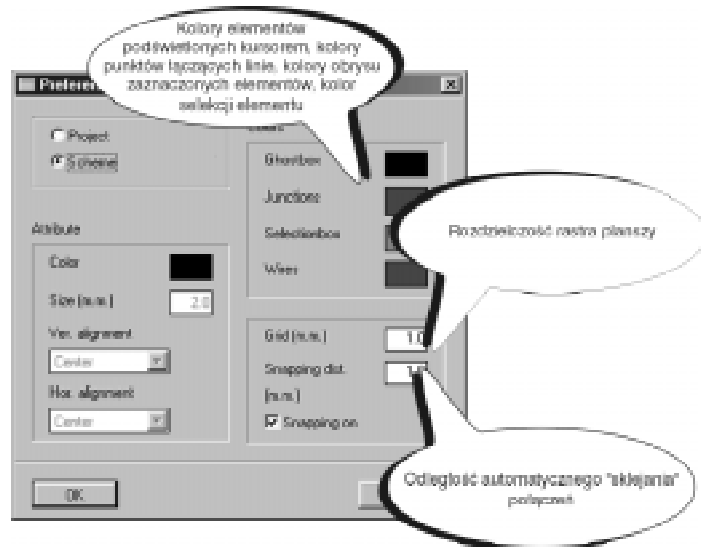
- *Auto wire option* umożliwia ustawienie parametrów automatycznego prowadzenia połączeń pomiędzy elementami bibliotecznymi podczas rysowania schematu (rys. 16).
- w *Configure toolbar* możemy dokonać samodzielnej konfiguracji paska narzędziowego, zaznaczając nazwę interesującego nas narzędzia i używając przycisku „->” dodać jego ikonę do prawego okna lub też używając przycisku „<-” usunąć ikonę narzędzia z paska (rys. 17).

Bardzo użytecznym poleceniem jest *Page layout options*, za pomocą któ-

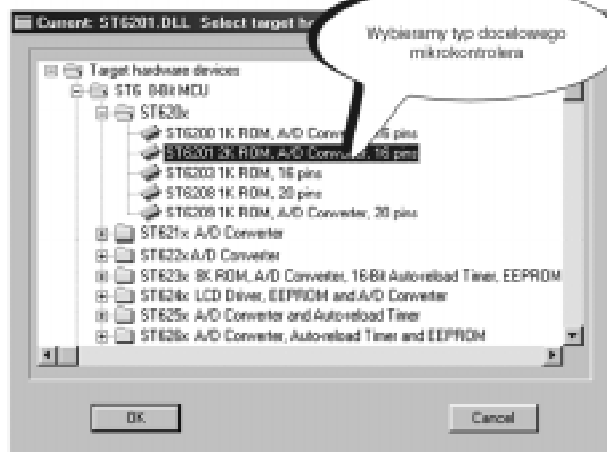


Rys.18.

	Wycięcie elementu
	Zapisz
	Otwórz schemat
	Nowy schemat
	Kopiuuj
	Wklej
	Analyszer
	Simulator
	Powiększenie ręczne
	Strzałka wyboru
	Łącznik elementów
	Komunikaty kompilatora
	Informacje analizera
	Informacje analizera
	Zmiana atrybutów
	Informacja o elemencie
	Lustro elementu
	Obrót elementu
	Tworzenie etykiety
	Element z biblioteki
	Otwórz bibliotekę
	Kopiuuj element
	Powiększ
	Pomniejsz
	Centruj schemat
	Widok całego projektu
	Pomoc



Rys. 19.



Rys. 20.

rego dokonujemy doboru parametrów planszy, na której będziemy rysować nasz schemat (rys. 18). Polecenie *Preferences* umożliwia między innymi dokonanie wyboru koloru linii łączących ze sobą elementy biblioteczne (rys. 19).

Po wybraniu opcji *Select hardware* otwiera się okno jak na rys. 20, w którym dokonujemy wyboru mikrokontrolera, dla którego będzie „rysowany” program. W naszym projekcie

użyjemy mikrokontrolera ST62T01, więc należy wybrać z listy *st6201.dll* i kliknąć OK.

Krzysztof Górski, AVT
krzysztof.gorski@ep.com.pl

Na płycie CD-EP2/2001B publikujemy ST6-Realizera w pełnej wersji funkcjonalnej. Jest on także dostępny (wraz z katalogiem procesorów ST62) na płycie CD-EP2.