

# ST6-Realizer w praktyce

## czyli zostać "malarzem" programów - kontroler napięcia akumulatora samochodowego, część 2

W drugiej części artykułu pokażemy dwa ostatnie etapy realizacji przygotowywanego projektu: jego symulację oraz programowanie procesora. W ten sposób kończymy opis - prawda, że szybko? - pełnego cyklu „pisania” programu dla prostego urządzenia. Opracowanie tego programu dowolną ze standardowych metod wymagałoby znacznie więcej trudu.

W pierwszej części artykułu przedstawiliśmy trzy pierwsze etapy realizacji projektu, które kończą się kompilacją przygotowanego programu. Zgodnie z prawami Murphy'ego efekty uzyskane po skompilowaniu programu są rzadko zgodne z zamiarami programisty i dobrym zwyczajem jest analiza jego działania. ST6-Realizer jest standardowo wyposażony w doskonały analizator, którego sposób wykorzystania przedstawiamy w artykule. Rozpoczynamy więc!

### Etap 4 - weryfikujemy sposób działania programu

Program symulatora można uruchomić bezpośrednio z paska narzędziowego ST6-Realizera (rys. 7). Po otwarciu się okna ST6-Simulator należy wybrać w menu *File/New* i wskazać plik zawierający przygotowany uprzednio schemat symulowanego projektu (rys. 8). Jest on automatycznie przekształcany do pliku z rozszerzeniem \*.sef - rys. 9, w którym są przechowywane informacje o wszelkich nastawach symulacji. Program umożliwia nadanie dowolnej nazwy plikowi \*.sef, ale najrozsądniejszym wyjściem jest nazwanie go w taki sam sposób, jak pliku zawierającego schemat.

Ogromną zaletą programu symulacyjnego ST6-Realizera jest jego intuicyjna obsługa. Wyobraźmy sobie standardową, laboratoryjną procedurę testowania projektowanego przez nas urządzenia - niezbędny jest jakiś zasilacz generujący regulowane napięcie oraz przyrząd umożliwiający obserwację stanów logicznych na wyjściach mikrokontrolera. Wszystkie te elementy znajdziemy w ST6-Simulatorze!

W dość „tajemniczy” sposób rozwiązali twórcy ST6-Realizera sposób dodawania do schematu generatorów sygnałów oraz urządzeń monitorujących. Ich instalację należy rozpocząć od wskazania myszką połączenia, do którego chcemy wybrany moduł dołączyć (rys. 10). Wskazane

połączenie zostaje automatycznie otoczone ramką, a na pasku narzędziowym uaktywniane są ikony (rys. 11), pozwalające umieścić na planie schematu wybrany moduł.

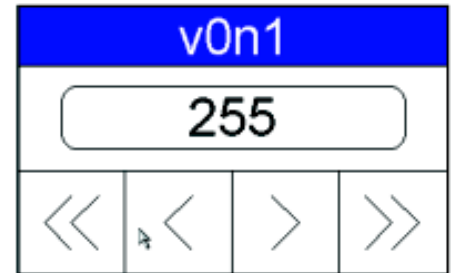
Dostępne są:  
 ✓ 8-bitowy programator numeryczny, przy pomocy którego można ręcznie zadać dowolną wartość z przedziału 0.255. Jego symbol graficzny przedsta-

wiono na rys. 12. Nie jest on wykorzystywany w symulacji prezentowanego projektu.



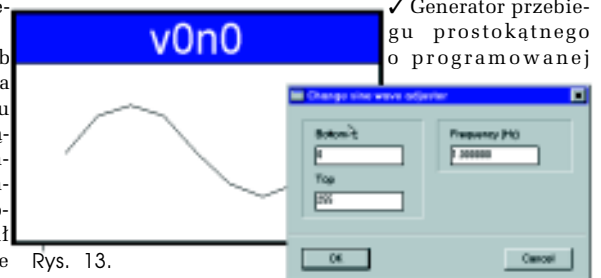
Rys. 11.

✓ Generator przebiegu sinusoidalnego o regulowanej amplitudzie, offsecie i częstotliwości. Zakres zmian amplitudy odpowiada 8-bitowej szynie danych. Graficzny symbol tego elementu przedstawiono na rys. 13. Obok widoczne jest okno kon-



Rys. 12.

figuracyjne, przy pomocy którego można w dowolny sposób ustalić parametry przebiegu. Element ten wykorzystamy w procesie symulacji, gdzie zastąpi napięcie akumulatora.



Rys. 13.

częstotliwości, wypełnieniu, amplitudzie i offsecie. Symbol graficzny tego elementu przedstawiono na rys. 14. Obok znajduje się okno konfiguracji, przy pomocy którego można dobrać parametry przebiegu wyjściowego. Maksymalna wartość generowana na wyjściu tego elementu wynosi 255, co odpowiada 8-bitowej szynie danych.

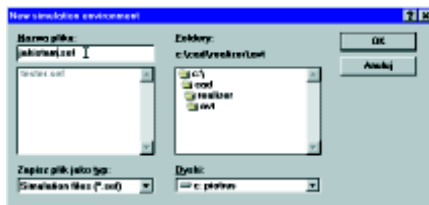
✓ Tabele opisujące zmiany wartości danych na 8-bitowej szynie. Symbol graficzny tego elementu przedstawiono na rys. 15. Jest to bardzo silne narzędzie symulacyjne, ponieważ umożliwia dokładną analizę czasową układu. W prawej części okna konfiguracyjnego (rys. 15) program tworzy uproszczony wykres czasowy przebiegu zadanego przez użytkownika.



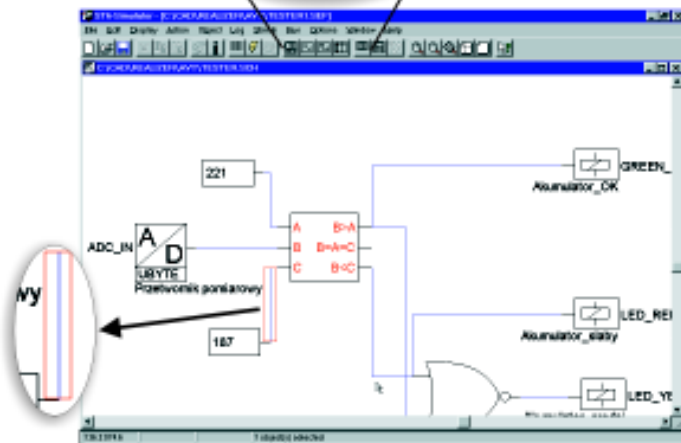
Rys. 7.



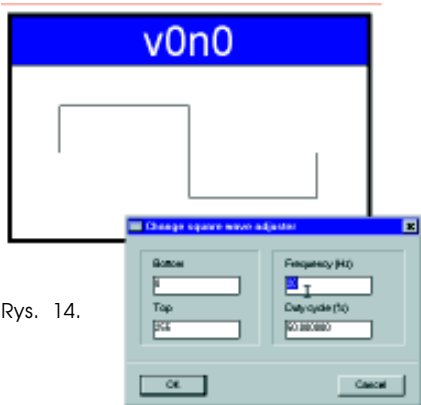
Rys. 8.



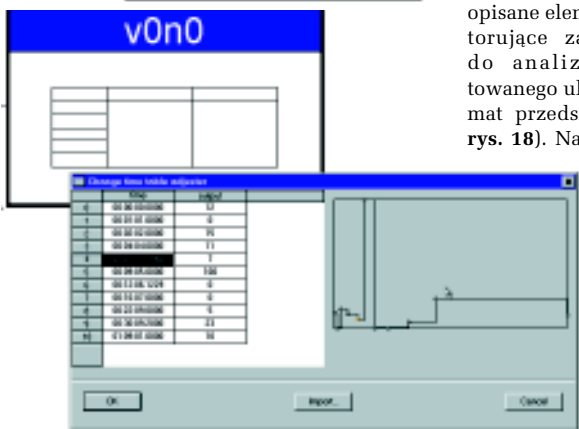
Rys. 9.



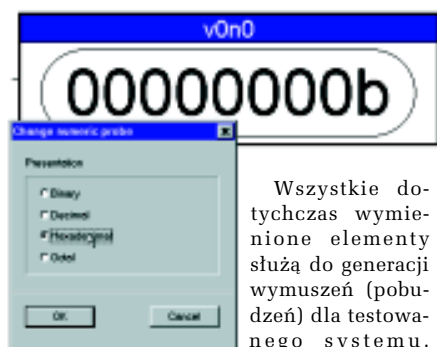
Rys. 10.



Rys. 14.



Rys. 15.

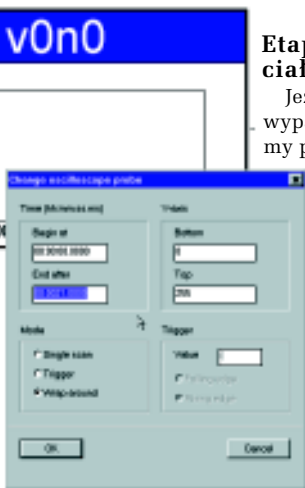


Rys. 16.

Wszystkie dotychczas wymienione elementy służą do generacji wymuszeń (pobudeń) dla testowanego systemu. Twórcy symulatora przewidzieli także dwa moduły umożliwiające obserwację stanów wyjść oraz stanu wejść.

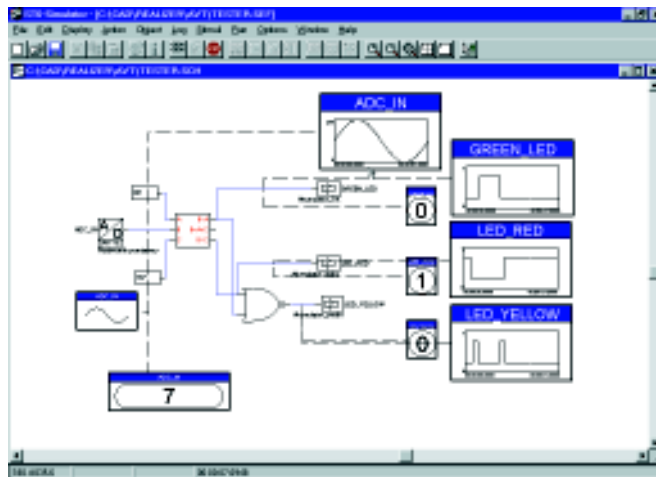
Role standardowego wskaźnika stanu szyny danych może spełniać sonda logiczna, której symbol graficzny oraz okno konfiguracyjne przedstawiono na rys. 16. Możliwe jest wyświetlanie wskaźników w jednym z czterech systemów liczbowych, a sonda automatycznie dopasowuje liczbę linii wejściowych (1..8) w zależności od szerokości monitorowanej szyny danych.

Drugim modulem wykorzystywanym do monitorowania stanu linii danych jest oscyloskop, którego symbol graficzny oraz okno konfiguracyjne przedstawiono na rys. 17.



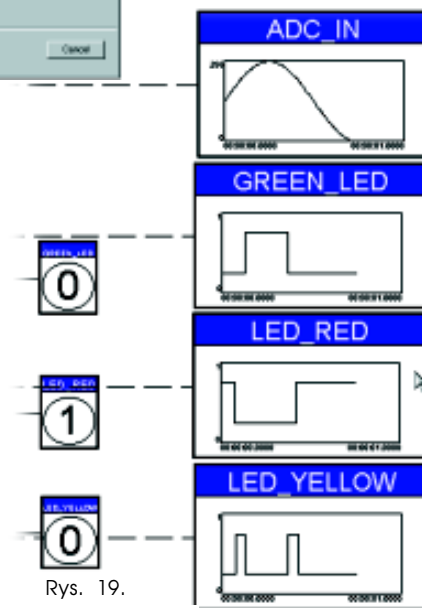
Rys. 17.

17. Oscyloskop może pracować w sposób zadany przez użytkownika - jednorazowo, w sposób ciągły oraz wyzwalany przez wskazane zbocze wybranego sygnału pojawiającego się na szynie danych. Możliwe jest także ustalenie czułości oscyloskopu, co pozwala na precyzyjne śledzenie niewielkich oscylacji wokół stałej wartości na szynie danych. Obydwa opisane elementy monitorujące zastosowano do analizy projektowanego układu (schemat przedstawiono na rys. 18).



Rys. 18.

Na rys. 19 pokazane zostały efekty pracy monitora napięcia akumulatora samochodowego widoczne na ekranach „oscyloskopów” dołączonych do wejścia i wyjść mikrokontrolera.



Rys. 19.

### Etap 5 - abstrakcja ciałem się staje

Jeżeli analiza pracy układu wypadła pomyślnie, to możemy przejść do przedostatniego etapu realizacji projektu - tzn. zaprogramowania mikrokontrolera. Najtań-

szym sposobem jest wykorzystanie kitu AVT-363 (ew. z adapterem AVT-363A) oraz oprogramowania znajdującego się na płycie CD-EP2 (pomyślnie podaliśmy w EP10/98, że jest to płyta CD-EP4). W katalogu \sgs\_thom.st6\sk622xa1 znajduje się instalacyjna wersja oprogramowania do obsługi programatora. Wystarczy uruchomić program *setup.exe*, który podpowie kolejne kroki.

Po zainstalowaniu programu należy podłączyć programator do wybranego złącza LPT (1 lub 2) i skopiować do katalogu z zainstalowanym programem plik *kit622x.de\_*, zmieniając jego nazwę na *kit622x.dev*. Oryginalny plik konfiguracyjny zostanie w ten sposób usunięty, ale nie będzie on już potrzebny.

Następnie uruchamiamy program *st622x.bat*, który automatycznie inicjuje sterowniki niezbędne do działania właściwego programu. Oprogramowanie pracuje zarówno w DOS, jak i z Windows 3.1x/95/98. W niektórych przypadkach (dotyczy Windows 95/98) należy w pliku *config.sys* dopisać linię *switches = /c*, co likwiduje możliwość błędnej pracy programu.

Po uruchomieniu programu *st622x.bat* wybieramy w menu typ procesora, dla którego przygotowano projekt (rys. 20). Przy pomocy opcji *lop* w górnym menu ustalamy, do którego z portów LPT dołączono programator. Następnie należy załadować do bufora program w postaci pliku *HEX* (powstaje on w wyniku pracy ST6-Realizera) i zainicjować programowanie procesora (opcja *Prog*). Programowanie kończy komunikat o poprawnym lub błędnym wyniku programowania. W pierwszym przypadku - możemy przejść do ostatniego etapu projektu, tj. montażu i uruchomienia monitora.

**Piotr Zbysiński, AVT**

Program *ST6-Realizer* oraz komplet narzędzi i danych katalogowych do mikrokontrolerów *ST62* znajdują się na płycie *CD-EP2*.

*Uwaga! Prosimy o nadsyłanie pomysłów, które chcielibyście zrealizować przy pomocy ST6-Realizera. Wybrane najciekawsze propozycje przedstawimy na łamach EP.*

Komplet plików wchodzących w skład projektu *TESTER* znajduje się w Internecie pod adresem: [www.avt.com.pl/avt/ep/ftp](http://www.avt.com.pl/avt/ep/ftp).



Rys. 20.