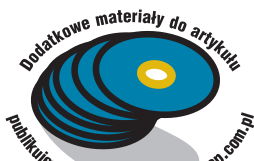


Rejestrator stanów logicznych dla PC

Nie wszyscy elektronicy mogą sobie pozwolić na zakup analizatora stanów logicznych. Napisany przeze mnie program, spełniający funkcję rejestratora sygnałów cyfrowych, może zastąpić te kosztowne przyrządy podczas prostych pomiarów.

Rekomendacje: szczególnie polecamy początkującym elektronikom, którzy minimalnym nakładem czasu i środków mogą wykonać bardzo przydatny i funkcjonalny rejestrator sygnałów cyfrowych.



Rejestrator-analizator stanów logicznych można wykonać na wiele sposobów, ale bez wątplenia najłatwiejszy to powierzyć tę rolę komputerowi PC. Rozwiązanie takie ma wadę, ponieważ nie jest możliwe precyzyjne odmierzenie czasu próbkowania, ale walory użytkowe takiego rejestratora w znacznym stopniu ją niwelują. Analizowane sygnały są wprowadzane do PC poprzez równoległy port drukarkowy (Centronics). W podstawowej wersji tego interfejsu mamy do dyspozycji 5 wejść, których stany możemy odczytać są to sygnały BUSY, ACK, PAPEROUT, ONLINE oraz !ERROR. Aby nie uszkodzić linii interfejsu (choćby jej nieumyślnym dołączeniem do napięcia wyższego od 5 V), który jest dzisiaj zwykle zintegrowany z płytą główną komputera, należy go odpowiednio zabezpieczyć.

Linie współczesnych wersji Centronicsa są wewnętrznie „podciągnięte” do napięcia 3,3 V. Wystarczy więc zastosować najprostszą optoizolację transoptorową (stosując na przykład transoptory CNY17 lub 4N35), w której tranzystor będzie zwieriał linię wejściową portu do masy w momencie przepływu prądu przez diodę LED - rys. 1. Natężenie prądu płynącego przez tę diodę należy oczywiście ograniczyć, np. za pomocą rezystora.

Obsługa programu

Okno działającego programu (dla Windows 98 i systemów pochodnych) pokazano na rys. 2. Ustawiane w nim czasy należy traktować jako orientacyjne. W celu zwiększenia dokładności ich odmierzenia warto wyłączyć wszystkie inne aplikacje.

Program wyświetla orientacyjny wykres, który ma po-

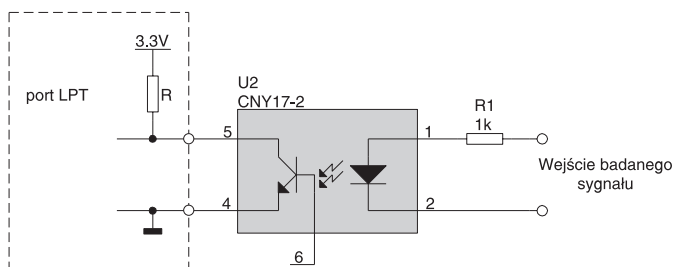
móc w analizowaniu przebiegów. Obsługa jest dość intuicyjna. Po kliknięciu na wykres możemy przesunąć jego zawartość za pomocą klawiszy kursorów: <, > oraz Home, PageUp, PageDown. Na pierwszym planie znajduje się przebieg sygnału podawanego na linię !ERROR, za nim („w głąb”) widać kolejno linie, którym odpowiada coraz wyższy numer bitu w odczytanym bajcie z rejestru BASE+1. Liczby pod słupkami wykresu oznaczają numer próbki (pomiaru). W ten sposób możemy np. określić czas trwania impulsów. Na wykresie mieści się maksymalnie 999 próbek, zaś cały bufor zawiera 65535. Zawartość bufora możemy eksportować, np. do Excela, co pozwala wykonać lepsze wykresy, analizę zgromadzonych danych itp. W tym celu wystarczy kliknąć na przycisk Zrzuci bufor do pliku. Wówczas w katalogu programu zostanie utworzony plik tekstowy buf_data.txt zawierający dane możliwe do zaimportowania do Excela jako plik tekstowy z separatorem w postaci spacji. Program zawsze zapisuje zawartość całego bufora, bez względu na to, ile w nim znajduje się próbek (poza obszarem próbek znajdujących się zera).

Przed dokonaniem pomiaru należy wybrać adres bazy portu LPT (BASE), który będzie wykorzystywany jako wejście danych rejestratora. Pomiar można przeprowadzić dwoma sposobami: bez synchronizacji oraz z synchronizacją sygnałem wyzwalającym. Niezależnie jednak od sposobu, należy wpiąć ustawić wartości Odstępów czaso-

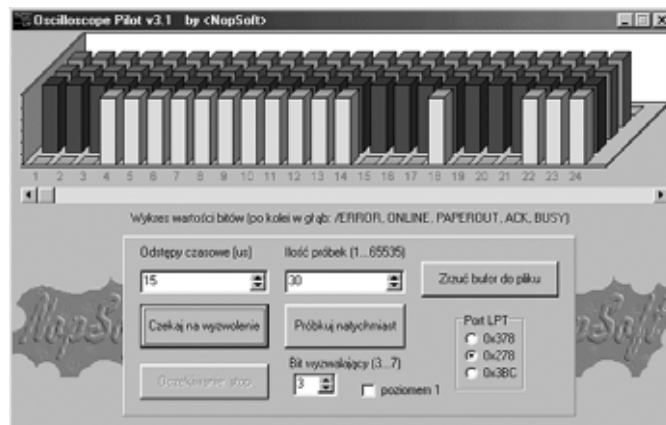
we oraz Ilość próbek. Pierwsza liczba określa odstęp czasu, po jakim ma być dokonywany pomiar. Druga zaś określa, ile razy zostanie przeprowadzony pomiar. Za ten czas samego pomiaru wyniesie minimum:

$$(ustalone\ odstepy\ czasowe + 4\ ms) * liczba\ próbek$$

Pomiar bez sygnału wyzwalającego jest przeprowadzany zaraz po wciśnięciu przycisku Próbki natychmiast. Pomiar z synchronizacją (która nie jest może dokładniejsza z powodu pewnych opóźnień, jednak zwykle wystarcza) wymaga ustawienia jeszcze dwóch parametrów. Pierwszym z nich jest Bit wyzwalający, który określa, z której linii pomiar ma być wyzwolany (np. bit 3 oznacza linię !ERROR). Drugim zaś jest poziom wyzwalający: czy wyzwolenie ma nastąpić poziomem 1 (high), czy 0 (low). Kiedy ustawimy powyższe parametry, wystarczy kliknąć na przycisk Czekaj na wyzwolenie. Jeśli teraz pojawi się odpowiedni sygnał na podanej linii, pomiar zostanie wykonany. W przeciwnym razie program będzie cały czas w stanie oczekiwania. Jeżeli będziemy chcieli przerwać ten stan, wystarczy przycisnąć aktywny przycisk Oczekiwanie: stop. Bez względu na to, czy pomiar zostanie dokonany, czy nie, zawartość wykresu i bufora ulegnie zmianie (o czym należy pamiętać). Jeśli będziemy chcieli określić stan linii synchronizującej czy innej na chwilę przed wyzwoleniem, możemy sygnał wyzwalający pozostawić bez zmian, a na odpowiedniej linii umieścić układ opóźnia-



Rys. 1. Schemat elektryczny optoizolowanego wejścia rejestratora



Rys. 2. Widok okna programu obsługującego rejestrator

Tab. 1. Przypisanie linii wejściowych do styków w typowych gniazdach i w rejestrze danych o adresie BASE+1

Bit portu BASE+1	7	6	5	4	3	-
Sygnal	BUSY	ACK	PAPER OUT	ONLINE	!ERROR	Masa
Numer styku w DB25	11	10	12	13	15	18...25 (wybór np. 21)
Numer styku w DB36	11	10	12	13	32	19...30, 33

jący (złożony np. z kilku bramek, obwodu RC lub RL).

W przypadku nowszych wersji programu (od 3.12) dostępna jest opcja *Czas jest krytyczny*, którą można wykorzystać, kiedy badamy szybkie przebiegi. Jednak należy ją wykorzystywać ostrożnie, gdyż istnieje sytuacja, w których doprowadzimy do zawieszenia się systemu, np. zbyt długi czas próbkowania

czy - w przypadku wyzwania - brak odpowiedniego sygnału synchronizującego. Oprócz tego dostępna jest opcja (od wersji 3.15) ukrycia okna programu na pasku zadań w tzw. *trayu*.

Instalacja programu

Aplikacja nie ma własnego instalatora. Jako że zostały użyte niestandardowe komponenty i kontrolki ActiveX,

wymagane są dodatkowe biblioteki z Delphi 3.0. Dlatego przed uruchomieniem programu należy skopiować plik *cfx32.ocx* do katalogu systemowego (najczęściej *c:\windows\system*), a następnie dodać do rejestru odpowiednie klasy, uruchamiając skrypt w pliku *cfx.reg*. Aplikacja ponadto wymaga sterownika DriverLINX, który można pobrać ze strony DiskDude:

- <http://venezia.cx/~diskdude/software/cbuilder/index.html>,
- <http://venezia.cx/~diskdude/files/cbuilder/DLPortIO/DLPortIO.zip>,
- <http://venezia.cx/~diskdude/files/cbuilder/DLPortIO/port95nt.exe>.

Powyższa czynność nie jest jednak konieczna w przypadku systemu Windows 98. Wystarczy do wspomnianego wyżej katalogu systemowego skopiować jeszcze pliki *dlportio.sys* oraz *dlportio.dll*. Wszystkie te czynności możemy wykonać automatycznie, uruchamiając plik wsadowy *syst_upd.bat*. W razie konieczności należy tylko uruchomić ponownie system, a następnie program *oscpil.exe*. Jeśli powyższe czynności nie przyniosą skutku, trzeba kolejno wykonać polecenia: *regsvr32 /u cfx32.ocx* oraz *regsvr32 cfx32.ocx*.

Janusz Użycki
nopssoft@poczta.onet.pl

Programator ISP mikrokontrolerów AVR z serii AT90S

*Pomimo zamieszczenia wywołanego przez ostrą selekcję przeprowadzoną niedawno wśród mikrokontrolerów AVR przez firmę Atmel, mikrokontrolery z serii AT90S są nadal produkowane (choć określane jako „mature“, czyli przewidziane do wycofania z produkcji), można je także nadal bez większego trudu kupić w sklepach elektronicznych. **Rekomendacje:** projekt polecany szczególnie fanom mikrokontrolerów AVR, którzy chcą wykonać sobie - praktycznie beznakładowo - funkcjonalny programator ISP.*

Autorem programu - bo prezentowany programator składa się przede wszystkim z programu - jest Słowak Juraj Lihosith. Program AVR Programmer współpracuje z bardzo prostym interfejsem (schemat elektryczny pokazano na rys. 1), za pomocą którego można programować następujące mikrokontrolery wyposażone w interfejs SPI: AT90S1200, AT90S2313, AT90S2323, AT90S2343, AT90S2333, AT90S4414, AT90S4434, AT90S4433, AT90S8515, AT90S8535 oraz mikrokontroler z rodziny '51 - AT89S8252. Dane można wprowadzać do pamięci programu Flash oraz danych EEPROM, możliwe jest także programowanie i kasowanie bitów zabezpieczających i konfiguracyjnych, do których dostęp podczas programowania ISP przewidział producent w oficjalnych specyfikacjach. Interfejs programujący jest dołączany do portu równoległego (Centronics) komputera PC. Dioda LED sygnalizuje aktywność programatora. Sygnały XTAL1,

SCK, MOSI, MISO, GND i RESET należy dołączyć do tak samo oznaczonych wyprowadzeń wymienionych mikrokontrolerów, a linia VCC służy głównie do zasilania diody LED. Interfejs może zostać nieco uproszczony i wtedy nie są wykorzystywane linie VCC i XTAL1 - szczegóły można znaleźć w pomocy programu AVR Programmer.

Na rys. 2 pokazano widok okna działającego programu, który, jakkolwiek niezbyt duży, wymaga instalacji. Podczas korzystania z programu należy pamiętać o zweryfikowaniu działania programatora dla różnych prędkości transmisji (*Interface Speed*) - w większości testowanych przypadków najlepszą stabilność pracy uzyskiwano dla prędkości *Medium*. W zależności od wybranej konfiguracji pracy interfejsu pokazane-



go na rys. 1, trzeba także ustalić odpowiedni tryb jego pracy (*Interface Mode*).

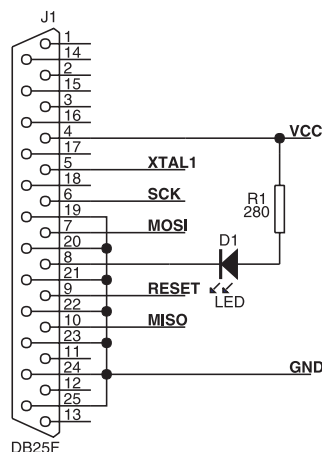
Autor programu udostępnia go bezpłatnie, ale przewiduje możliwość jego rejestracji za równowartość 3 USD. Rejestracja nie zwiększa funkcjonalności programu.

TG

Więcej informacji o programie prezentowanym w artykule można znaleźć w Internecie pod adresem: <http://avr.webpark.sk/>.

Instalacyjną wersję programu AVR Programmer publikujemy na CD-EP8/2003B, jest ona dostępna także w Internecie pod adresem: <http://avr.webpark.sk/Files/setup.exe>.

Na płycie CD-EP8/2003B zamieściliśmy program AVR Programmer 3.3 (pracuje z Windows 95/98/Me) oraz noty katalogowe mikrokontrolerów obsługiwanych przez niego (oprócz AT90S4414).



Rys. 1. Schemat elektryczny programatora



Rys. 2. Wygląd okna programu sterującego