

ST6-Realizer

część 6



Narysuj swój program!

Sterowanie wyświetlaczem alfanumerycznym LCD

Mogłoby się wydawać, że narysowanie odpowiedniego schematu działania za pomocą ST6-Realizera jest trudne. Wielu użytkowników programu wręcz twierdziło, że jest to niemożliwe. Jednak ST6-Realizer, przy odrobinie fantazji, staje się mocnym i szybkim narzędziem projektowym, umożliwiającym w sposób nieskomplikowany stworzenie odpowiedniego programu obsługi wyświetlacza. Do naszych eksperymentów należy wykonać prosty układzik, którego schemat elektryczny pokazano na **rys. 1**. Wyniki naszych prac oglądać będziemy na dowolnym wyświetlaczu LCD1x16 ze standardowym sterownikiem HD44780.

Nie będę tu opisywał dokładnie zasady działania wyświetlacza i wszystkich jego procedur. Wiele już na ten temat napisano na łamach Elektroniki Praktycznej i Elektroniki dla Wszystkich. Zainteresowanym szczególnie polecam cykl artykułów pt. „Alfanumeryczne wyświetlacze LCD“ (m.in. EdW11/97).

Jak to zrobić?

Do komunikacji pomiędzy mikrokontrolerem a wyświetlaczem służą trzy linie sterujące (RW, RS, E) oraz osiem linii danych (D0..7). Wyświetlacz może być sterowany poprzez 8- lub 4-bitową magistralę danych. Przy zastosowaniu mikrokontrolera ST62T10/20, z niewielką liczbą wyprowadzeń, ośmiobitowe sterowanie

wyświetlaczem jest wykluczone. Dlatego zajmiemy się trudniejszym do wykonania czterobitowym sterowaniem wyświetlacza LCD, gdzie używane są linie danych D4..D7 i linie sterujące RW, RS, E. Przełączenie sterownika wyświetlacza w tryb czterobitowy odbywa się podczas jego inicjalizacji po włączeniu zasilania. Przesyłanie danych pomiędzy sterownikiem i sterującym jego pracą mikrokontrolerem odbywa się zawsze dwuetapowo: najpierw jest przesyłany starszy półbajt instrukcji lub danej, a następnie młodszy.

Ponieważ w większości przypadków informacje są wpisywane do sterownika wyświetlacza, a nie odczytywane z niego, w prezentowanym projekcie sygnał określający kierunek przesyłania danych pomiędzy sterownikiem i mikrokontrolerem został pominięty jako nieistotny.

Działanie programu

Na **rys. 2** przedstawiono przykładowy schemat programu obsługującego wyświetlacz LCD ze sterowaniem 4-bitowym. Każde naciśnięcie przycisku START powoduje przejście programu do kolejnego stanu odpowiadającego wyświetleniu wcześniej zadanych napisów, przez wygenerowanie odpowiednich sekwencji stanów logicznych na wejściach sterujących RS, E i danych przesyłanych do sterownika wyświetlacza D4..D7. Generator *osc LCD* generuje ciąg im-

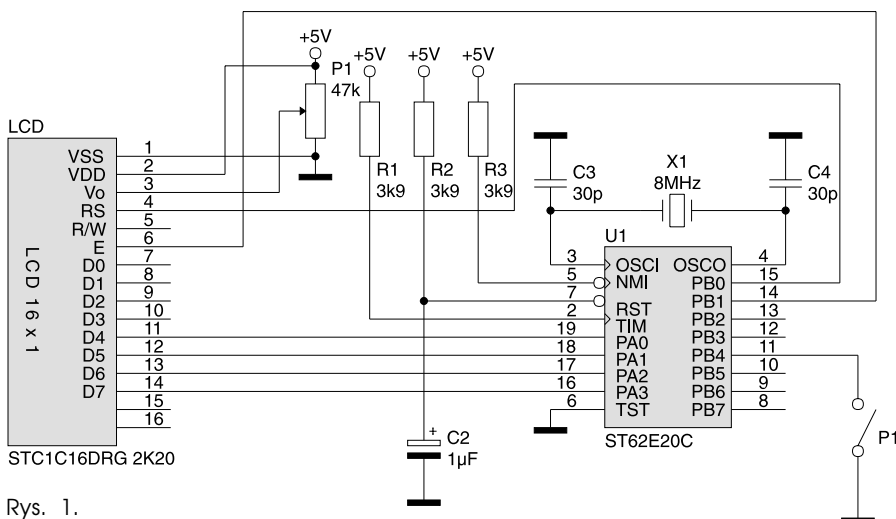
Z listów wynika, że wielu Czytelników zainteresowanych programem ST6-Realizer interesuje się także sposobem „ujarzmienia“ za jego pomocą alfanumerycznego wyświetlacza LCD. W ostatniej części tego krótkiego kursu pokażemy jeden z możliwych, sprawdzonych przez nas sposobów.

pulsów, które poprzez dwa niezależne multipleksery *mux1* podawane są na:

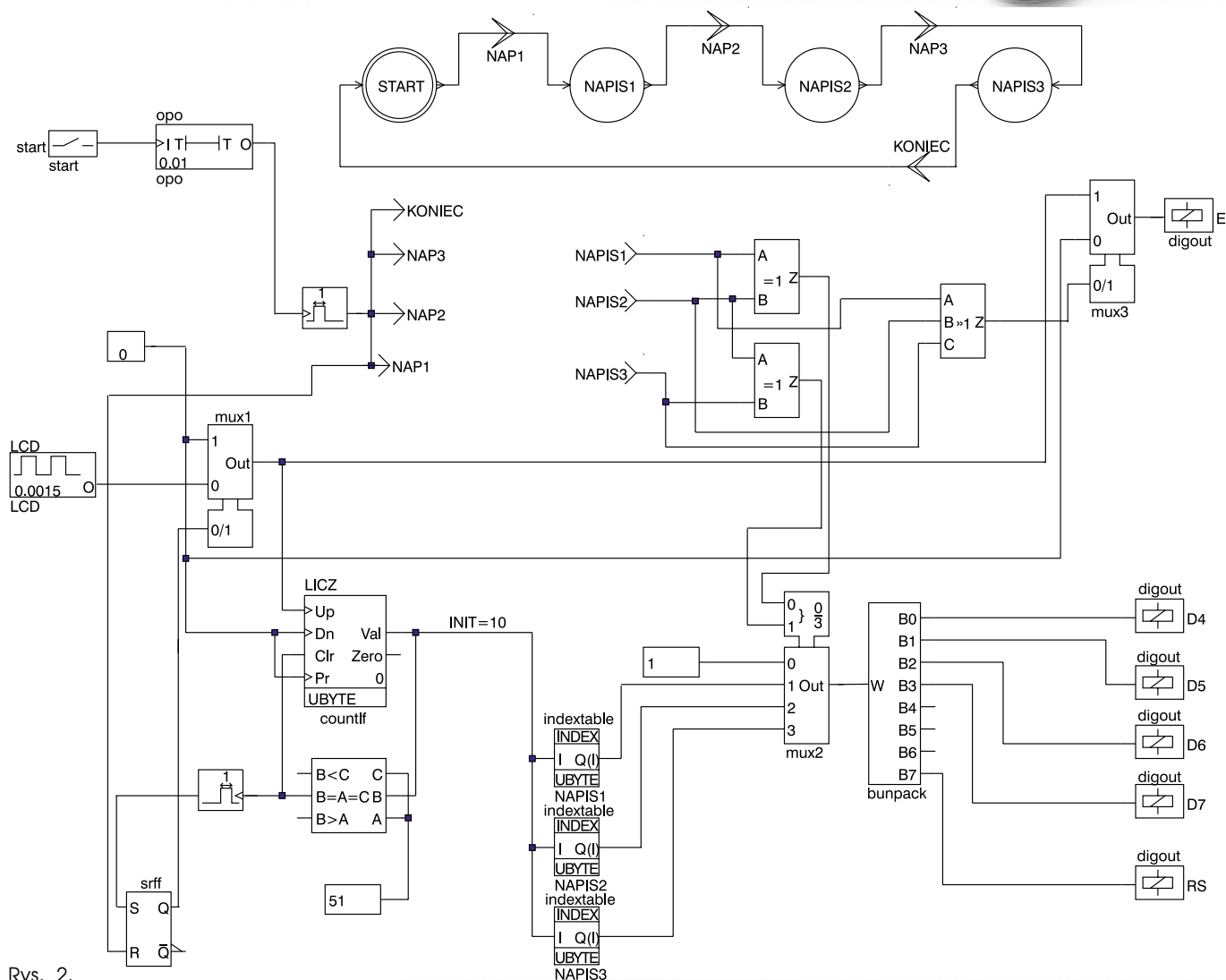
- wejście *Up* licznika *countlf*,
- wyjście cyfrowe *digout E*.

Wyjście *val* licznika *countlf* jest połączone z wejściem tablic *indextable*. Licznik zlicza impulsy pochodzące z *osc LCD* do wartości 51, a następnie zostaje wyzerowany. Dzieje się to w wyniku porównania przez komparator *comp* wartości na wyjściu licznika z wartością stałą. Na wyjściu $B=A=C$ komparatora pojawia się wysoki poziom napięcia, które podane na wejście *Clr* licznika zeruje go. Komparator również steruje pracą przerzutnika *srff*. Zamiast komparatora możemy użyć tablicy *lookuptable* o zawartości *Def.0 ; 51,1*. Po podaniu wysokiego poziomu napięcia na wejście ustawiające S przerzutnika *srff*, na jego wyjściu Q pojawia się poziom wysoki, w wyniku czego zostaje zablokowany generator *osc*. Brak impulsów na wejściu zliczającym *up* licznika powoduje, że licznik przestaje zliczać. Odblokowanie licznika i powtórzenie całej opisananej sekwencji zliczania nastąpi po kolejnym naciśnięciu przycisku *START*. Takie rozwiązanie pozwala wygenerować przez licznik ciąg wartości od 0 do 51, co jest potrzebne do wpisania 16 znaków do rejestru wyświetlacza LCD. Cała procedura inicjalizacji wyświetlacza i następnie wpisywania znaków jest zawarta w tablicach *indextable*.

Wyjścia tablic *NAPIS1..3* połączone są poprzez multipleksor *mux2*, za pomocą którego jest wybierany komunikat do wyświetlenia, z wejściem *W* bloku *bunpack*. Sygnały z wyprowa-



Rys. 1.



Rys. 2.

dzeń bloku *bunpack* sterują modulem wyświetlacza poprzez wyjścia cyfrowe *digout* skonfigurowane jako *push-pull output*. Od zawartości tabel zależy, w jaki sposób i co będzie się ukazywać na wyświetlaczu LCD. W naszym przypadku będzie to 50 liczb, które są odpowiednikami sygnałów sterujących i danych.

Jak już wspomniano, wyświetlacz po włączeniu zasilania musi przejść proces inicjalizacji. Na *list. 1* przedstawiono zawartość pierwszych czterech pozycji w tabeli *NAPIS1* wraz z opisem. Zawierają one ciąg znaków niezbędnych do poprawnej inicjalizacji i skonfigurowania sterownika wyświetlacza LCD.

Kolejne 36 pozycji tabeli dotyczy wpisania tekstu do wyświetlacza (*list. 2*). Efektem przepisania zawartości tablic do pamięci i rejestrów sterownika LCD jest wyświetlenie napisu o treści „www.ep.com.pl”.

Wprowadzanie kolejnych dwóch napisów do wyświetlacza wygląda podobnie - pierwsza część to inicja-



List. 1. Kolejne słowa danych umożliwiające konfigurowanie sterownika wyświetlacza LCD (cyfry oznaczają wartość bajtów zapisanych dziesiętnie).

```
0
0
3 Function Set
3 Function Set
3 Function Set
2 Function Set - interfejs
czterobitowy
2 Function Set - liczba
wyświetlanych linii 2, format znaku
5x7
8
0 Display OFF
8
0 Display ON
1
0 Entry mode set - kierunek przesuwu
kursora (zwiększenie adresu
następnego znaku o 1)
6 Function Set - przesuwanie napisu
podczas wprowadzania wyłączone
```

lizacja, druga to wpisanie treści komunikatu.

Czytelnicy znający obsługę wyświetlaczy mogą zadać pytanie: po co za każdym razem przeprowadzać inicjalizację? Oczywiście jest to niepożądane, wystarczy zrobić to raz, a następnie zmieniać tylko wpisywany tekst lub też adresować poszczególne pozycje na wyświetlaczu i zmieniać znaki. Zastosowane rozwiązanie umożliwiło uproszczenie programu (schematu) do tego stopnia, że jest on czytelny i bardziej zrozumiały dla mniej zaawansowanych czytelników.

Sposób tworzenia tablic ze wzorcami wyświetlanych tekstów

Zakładam, że wszyscy „kursanci” znają choć trochę standardowe sposoby obsługi wyświetlaczy LCD. Jak zapewne wiecie, wszystkie znaki alfabetu łacińskiego, cyfry, znaki specjalne są ponumerowane według kodu ASCII, w którym znakowi odpo-

wiada liczba 8-bitowa. Przykładowo, literze „R” przyporządkowano kod 82 (zapisany dziesiętnie), czyli binarnie 01010010.

Jak zatem wpisać liczbę 82 do tablic *NAPIS1..3* wykorzystanych w schemacie programu? W przypadku zastosowania pomiędzy mikrokontrolerem i wyświetlaczem transmisji ośmiobitowej w tablicy możemy podać bezpośrednio liczbę dziesiętną. W przypadku zastosowania transmisji 4-bitowej, każdy wyświetlany znak zajmie dwa miejsca w tablicy (po jednym dla starszego i młodszego półbajtu). Tak więc chcąc do tablic *NAPIS1..3* wpisać dane pozwalające wyświetlić literę „R” w trybie sterowania 4-bitowego, musimy do wartości dziesiętnych odpowiadających półbajtom

0101 - 5 (starszy)

0010 - 2 (młodszy)

dodać liczbę 128. Wartość ta jest niezbędna, aby po podaniu na wejście *W* bloku *bunpack* uzyskać dodatkowo na wyjściu *B7* bloku *bunpack* poziom wysoki odpowiadający sygnałowi sterującemu RS. Po dodaniu tej liczby otrzymujemy wartości 133 i 130, które wpisujemy w odpowiednie miejsca tablicy. Podobnie postępujemy z resztą liter, cyfr i innych znaków, które mamy zamiar wyświetlać.

Mam nadzieję, że ta część kursu rozwieje część wątpliwości na temat sterowania wyświetlaczem i rzuci nowe światło na możliwości ST6-Realizera.

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

Nota katalogowa układu HD44870 oraz pliki źródłowe programu do projektu prezentowanego w artykule znajdują się na płycie CD-EP7/2001 w katalogu \noty katalogowe do projektow\kurs.

List. 2. Zawartość tablicy wyświetlającej napis (cyfry oznaczają wartość bajtów zapisanych dziesiętnie)

```
135
135 w
135
135 w
135
135 w
130
142 . kropka
134
133 e
135
128 p
130
142 . kropka
134
131 c
12 Instrukcja podająca adres 40 w
pamięci DD RAM pierwsza pozycja
liter w drugiej linii
0
134
143 o
134
141 m
130
142 . kropka
135
128 p
134
140 l
138
128 spacja
138
128 spacja
138
128 spacja
0
2 ustawienie kursora na pozycji
początkowej
0
```

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