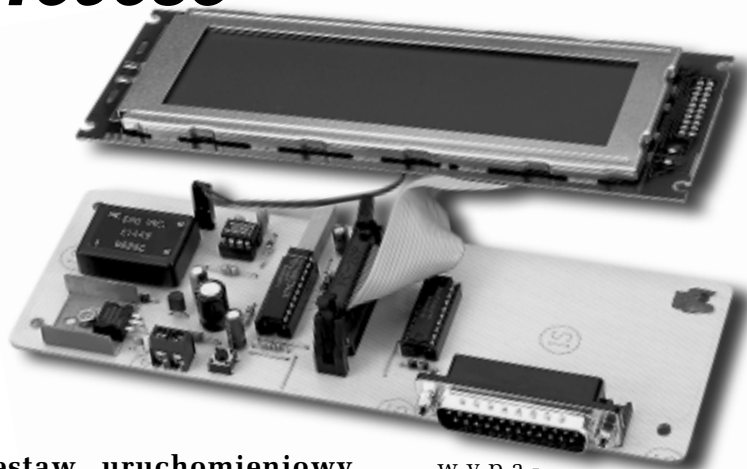


Programowanie sterowników wyświetlaczy graficznych, część 2

Wyświetlacz graficzny LCD z kontrolerem T6963C

W drugiej części artykułu prezentujemy zestaw uruchomieniowy oraz podstawowe funkcje oprogramowania przygotowanego do sterowania pracą graficznego wyświetlacza LCD i tworzenia plików graficznych w postaci binarnej. Zaczynamy od opisu zestawu uruchomieniowego.



Zestaw uruchomieniowy

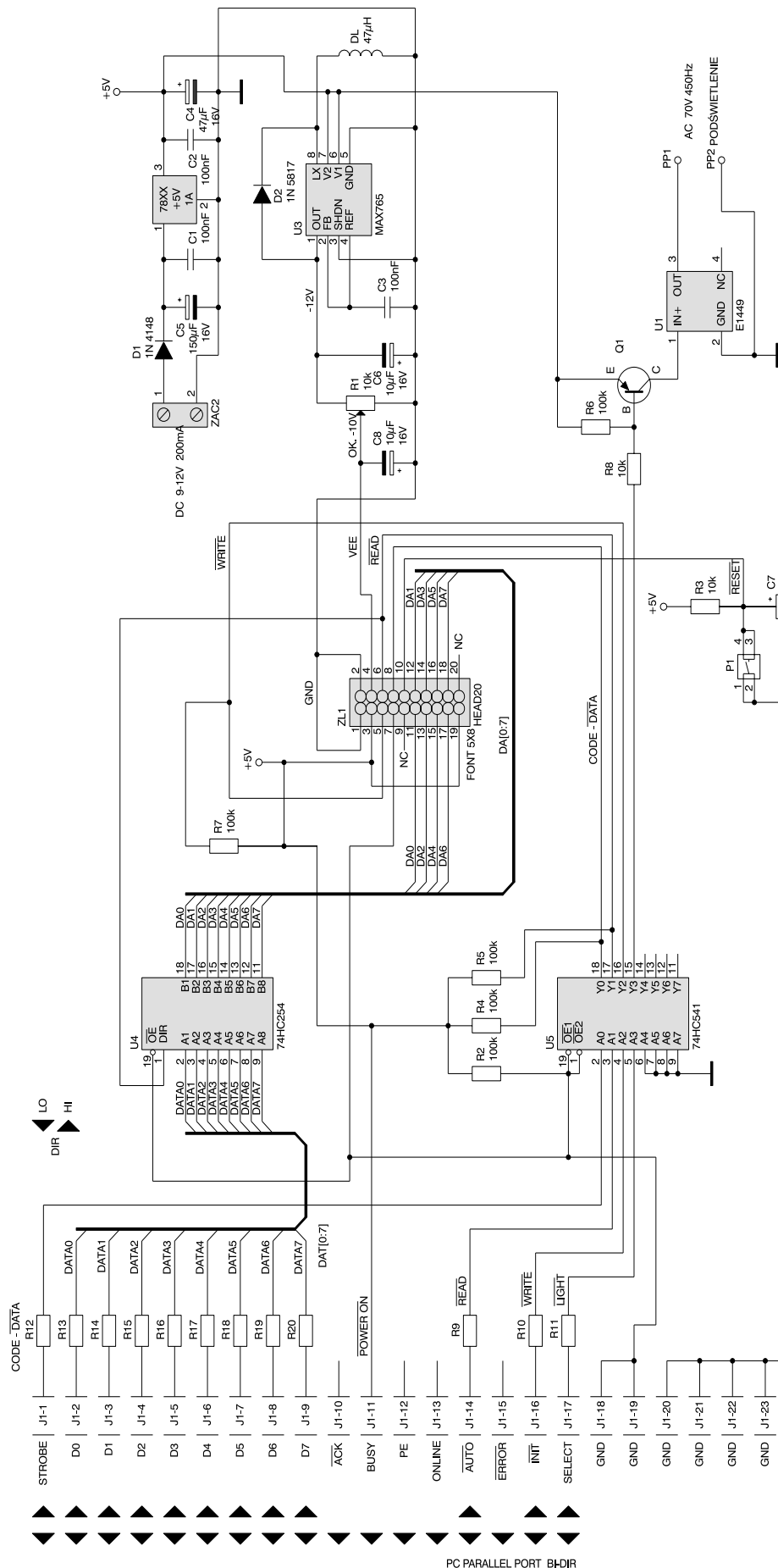
Cel skonstruowania zestawu uruchomieniowego był dwójaki:

- zapoznanie się z różnorodnymi możliwościami wyświetlacza w wygodny sposób (bez prowizorycznych podłączeń, które narażają drogi zespół na uszkodzenie oraz z prostym w obsłudze interfejsem sterującym);
- wykorzystanie do sterowania wyświetlaczem portu równoległego z poziomu Windows.

W efekcie powstała płytką połączeniową współpracująca z programem sterującym uruchamianym w środowisku Win95/98. Schemat przystawki jest przedstawiony na **rys. 6**. Na płytce (**rys. 7**) umieszczono elementy interfejsu łączącego (złącza, bufony), zasilania +5V i -12V oraz przetwornicy podświetlenia. Do połączenia z portem równoległym komputera (musi to być port pracujący w trybie bi-directional, inaczej byte-mode) służy gniazdo DB25HM. Potrzebny jest zatem kabel z końcówkami DB25M i DB25F. Jest to pewne utrudnienie, ale można do płytki interfejsu łatwiej przyłączyć inne sterowniki. Wprowadź linie sterujące wyświetlacza są elektrycznie zgodne z liniami portu, jednak na wszelki

wypadek (dla dodatkowego zabezpieczenia zarówno portu jak i wyświetlacza) zastosowano bufony: jednokierunkowy U5 - 74HC541 dla sterowania oraz dwukierunkowy U4 - 74HC245 dla danych. Przypisanie pinów portu sygnałom sterującym nie ma jakiegoś szczególnego znaczenia - zostało wybrane na etapie projektowania.

Oprócz opisanych wcześniej sygnałów sterujących wyświetlacza występuje jeszcze linia włączania podświetlenia (SELECT) oraz wejście kontroli włączania wyświetlacza (BUSY). Bufory są uaktywniane po połączeniu płytki kablem z komputerem (piny 18 i 19 złącza), co zabezpiecza je przed stanami nieustalonymi (mogącymi powodować konflikty pomiędzy buforami a wyświetlaczem). Połączenie z wyświetlaczem zapewnia 20-pinowe złącze ZL1. Można zastosować złącze dowolnego typu (nawet odcinek podwójnej listwy) - w prototypie zastosowano pionowe z "łapkami", znacznie ułatwiające demontaż zestawu. Elementy R3, C7 zapewniają zerowanie wyświetlacza. W trakcie uruchamiania przydaje się czasem też przycisk zerowania ręcznego P1. Wej-



Rys. 6. Schemat elektryczny interfejsu.

ście CE jest uaktywnione na stałe, na stałe też jest ustawiony rozmiar fontu 6x8 (wygląd fontów 8x8 nie jest zbyt rewelacyjny).

Blok głównego zasilania +5V zrealizowano w sposób tradycyjny, to jest ze stabilizatorem ST1 typu 7805 (zmontowanym na niewielkim radiatorze), pojemnościami filtrującymi oraz diodą zabezpieczającą D1. Taki układ pozwala na zasilanie płytki napięciami z zakresu 8..15V. Do uzyskania -12V użyto scalonej przetwornicy DC/DC Maxima MAX765 (U3). Ma to na celu zachowanie jednonapięciowego zasilania - równie dobrze można doprowadzić -12V z zewnątrz. Z racji małego poboru prądu, aplikacja MAX765 nie jest zoptymalizowana: dławik i kondensator zostały po prostu wzięte z podręcznych zapasów (normalnie dławik powinien być dobrany pod kątem przeniesienia odpowiednio dużego prądu, a elektrolit - dla minimalnego ESR).

Potencjometr wieloobrotowy R1 wyprowadzony na przód płytki służy do regulacji kontrastu wyświetlacza. Elementy R6, R8, Q1, U1 dostarczają napięcia przemiennego (ok. 70V i 400..450Hz) do zasilania podświetlenia. Zastosowano gotowe, dedykowane elementy: folię podświetlającą LFOB 08 (światło białe) i przetwornicę E1449 (ERG Inc.). Przetwornica może być zasilana napięciem z przedziału 3,0..5,5V, pobierając typowo prąd około 50mA. Można też użyć innych przetwornic - np. z oferty ELFY.

Rozmiary płytki są dostosowane do umieszczenia na niej - za pomocą dwóch skośnych wsporników - modułu wyświetlacza pod dogodnym kątem (w prototypie wsporniki są wykonane z plexi 10mm). Dla wygody użytkownika, od dołu przymocowane są nóżki.

Rezystory R9, R20 o wartości 68Ω zostały dolutowane w trakcie uruchamiania dla redukcji zakłóceń - kiedy układ nie bardzo chciał zadziałać. Chociaż ostatecznie okazało się, że przyczyna jest inna - na wszelki wypadek je pozostawiono.

Wstępne uruchomienie interfejsu sprowadza się do kontroli napięć zasilania oraz ustawienia potencjometrem R1 napięcia Vee ok. -9,5V. To w zasadzie przy starannym mon-

tażu powinno wystarczyć (przy budowie - dla uniknięcia niespodzianek - sprawdzano oddzielnie pracę każdej linii sygnałowej z użyciem pomocniczego oprogramowania). Przetwornicę podświetlenia włączamy raczej tylko z obciążeniem. Wyświetlacz dołączamy do interfejsu odcinkiem taśmy z zaciśniętymi wtykami (sygnały, zasilanie) oraz przewodem z 2-pinową wsuwką (podświetlenie) i mocujemy do wsporników z użyciem tulejek.

Konfiguracja i uruchomienie oprogramowania

Gotowy interfejs dołączamy opisanym wcześniej kablem do wybranego portu równoległego komputera - zazwyczaj będzie to LPT1 z adresem \$378. Port należy ustawić w tryb bi-directional (byte-mode). W starszych rozwiązaniach może być wymagane przedstawienie zworek na karcie, nowsze płyty mają port wbudowany i wystarczy ustawienie trybu w BIOS-ie. Nie należy uruchamiać oprogramowania w trybach ECP/EPP, które używają sygnałów sterujących dla potrzeb wbudowanych mechanizmów handshakingu (nie uzyskamy łączności z interfejsem, a sam port może się zawiesić - przypadkowe uruchomienie w ECP na HP Vectra spowodowało utratę dwustronnej komunikacji z drukarką aż do całkowitego wyłączenia zasilania komputera). Natomiast w trybie standard doprowadzimy do konfliktu na liniach - zwarcia rzędu 45 mA nie powinny wprawdzie uśmiercić sprzętu, ale lepiej nie ryzykować (zwłaszcza z portem na płycie).

Następnym wymogiem jest uruchomienie systemu Windows 95/98. Program wykorzystuje bowiem bezpośrednio odwołania do portów I/O, które nie będą tolerowane przez Windows NT (w tym przypadku należy zrealizować dostęp do portów z użyciem dodatkowych komponentów).

Aplikację uruchamiamy z parametrem określającym numer portu. Cyfra „1” - oznacza LPT1 z adresem \$378, a „2” - LPT2 z adresem \$278. Dla prostoty zrezygnowano z automatycznej lokalizacji portów - takie przypisanie jest dzisiaj już praktycznie standardowe (w razie wątpliwości należy sprawdzić konfigurację

sprzętu). I uwaga - jako domyślny (bez parametru) jest przyjmowany LPT2 (na takim uruchamiano program). Działanie zostało przetestowane na konfiguracjach:

- Optimus Pentium 200MHz, z LPT2 na karcie ISA TC-210;
- HP Vectra Pentium MMX 200MHz, z LPT1 na płycie głównej.

W razie kłopotów z komunikacją pojawi się odpowiednie okienko ostrzegawcze. Szybkość działania programu sterującego nie może przekraczać podanych wcześniej wymogów czasowych kontrolera wyświetlacza. Spójrzmy na kluczowe procedury odczytu i zapisu (Object Pascal):

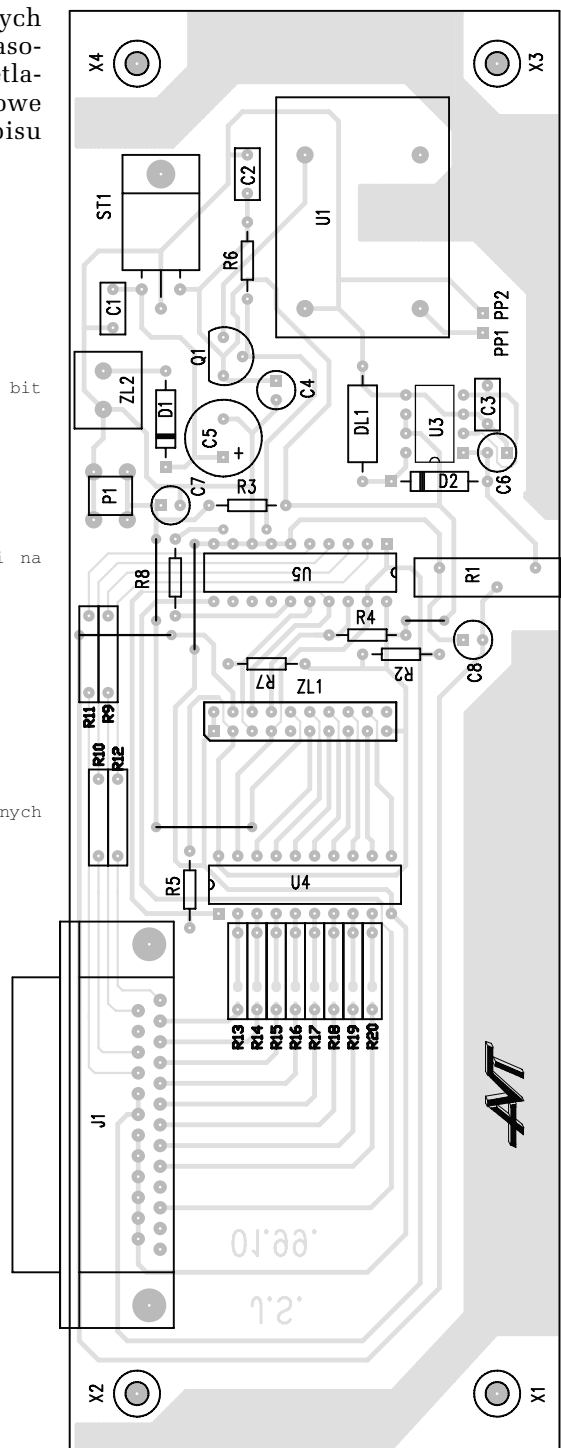
```
function ReadByte:Byte;
var
  W:Word;
  B:byte;
begin
  W:=LptBase+2;
  B:=BaseCtrl or $22;
  {linia Read = AUTOFEED - bit
  1[inwertowany] }
  asm
    PUSH DX
    MOV DX,W
    MOV AL,B
    OUT DX,AL { poziom niski na
    linii Read, DIR = In }
  POP DX
end;

W:=LptBase;
asm
  PUSH DX
  MOV DX,W
  IN AL,DX { dane z portu danych }
MOV B,AL
POP DX
end;

Result:=B;
W:=LptBase+2;
B:=BaseCtrl;
asm
  PUSH DX
  MOV DX,W
  MOV AL,B
  OUT DX,AL { poziom wysoki
  na linii Read, DIR = Out }
  POP DX
end;
end;
{-----}
procedure WriteByte(ABYTE:Byte);
var
  W:Word;
  B:Byte;
begin
```

```
W:=LptBase;
B:=AByte;
asm
  PUSH DX
  MOV DX,W
  MOV AL,B
  OUT DX,AL { dane do portu danych }
  POP DX
end;

W:=LptBase+2;
B:=BaseCtrl and not $4;
```



Rys. 7. Rozmieszczenie elementów na płycie drukowanej.

```

{linia Write = INIT - bit 2|nie inwer-
towany }
asm
PUSH DX
MOV DX,W
MOV AL,B
OUT DX,AL { poziom niski na linii
Write }
POP DX
end;

W:=LptBase+2;
B:=BaseCtrl;
asm
PUSH DX
MOV DX,W
MOV AL,B
OUT DX,AL { poziom wysoki na linii
Write - wpis }
POP DX
end;

end;

```

O ile nasz port będzie zbyt szybki może się okazać konieczne wprowadzenie odpowiednich opóźnień. Dla podanych powyżej konfiguracji konieczne było przy zapisie wcześniejsze wystawienie danych (przed opadającym zboczem WR). Okazało się, że zbyt szybko była wykonywana funkcja sprawdzająca status (ciągła pętla ReadByte) - ostateczna wersja jest podana poniżej (dla trybu zwykłego). Wywołania *Application.ProcessMessages* oprócz spowolnienia pętli pozwalają na wycofanie się z funkcji przy braku właściwego rezultatu (program nadal reaguje na komunikaty systemowe):

```

function CheckStatus:Boolean;
var
k:Integer;
B:Byte;
begin
k:=0;
SetCmdLine(COMMAND);
repeat
Application.ProcessMessages;
B:=ReadByte;
Inc(k);
if (k=100) then begin
Application.MessageBox('Sprawdź pod-
łączenia i|zasilanie',
'Brak odpowiedzi',mb_Ok);
Result:=False;
Exit;
end;
until ((B and $3)=$3);
Result:=True;
end;

```

Na tych elementarnych procedurach bazują pozostałe, bardziej złożone funkcje realizujące

opisane wcześniej operacje obsługi wyświetlacza. Natomiast część „ekranowa” programu pozwala na wygodne wprowadzanie odpowiednich komend. Są one zgrupowane w menu głównym:

- *Widok* - włączanie i wyłączenie trybu graficznego i tekstowego;
- *Tryb* - wybór stylu wyświetlania;
- *Kursor* - obsługa kursora (widoczność, wielkość, miganie i pozycja);
- *Znaki* - wybór generatora znaków;
- *Funkcje* - operacje dodatkowe: podświetlenie oraz programowe zerowanie wyświetlacza;
- *Program* - ma dwie części: przełącznik grafika/tekst odnosi się do zawartości ekranowego symulatora wyświetlacza, natomiast pozostałe komendy przesyłają odpowiednie dane z programu do RAM wyświetlacza;
- *Bmp* - umożliwia załadowanie bitmapy z pliku *.bmp.

Interfejs/symulator wyświetlacza ma dwa ekrany: graficzny, do którego ładujemy plik *.bmp oraz tekstowy, do którego możemy wpisywać własny tekst i ewentualnie nadawać mu atrybuty.

Ekran graficzny wyświetla bitmapę *.bmp w trybie „stretch”, tzn. dopasowuje ją do swojego wymiaru, gdy rzeczywiste wymiary bitmapy mogą być inne.

Wyświetlacz pokaże wymiary rzeczywiste, dlatego program informuje o załadowaniu bitmapy różnej od 240x64. Bitmapa z ekranu zostaje przekodowana i przesłana do wyświetlacza po wybraniu komendy *GraphMem* z menu *Program*.

Ekran tekstowy podzielony jest na siatkę znaków 40x8. Możemy w nią wpisywać tekst z klawiatury. Prawy przycisk myszy otwiera kontekstowe menu atrybutów. Wpisanie znaku oraz ustalenie atrybutu zawsze dotyczy całego zaznaczonego w siatce obszaru. Przesłanie tekstu oraz atrybutów do RAM jest realizowane komendami *TextMem* i *AttrMem* z menu *Program*.

Wybrano taką etapową obsługę dla zwiększenia pogładowości działania zestawu. Nic nie stoi na przeszkodzie, żeby w bardziej praktycznej aplikacji akcje klawiatury przesyłać natychmiast do wyświetlacza. Taką aplikacją

WYKAZ ELEMENTÓW

Interfejs

Rezystory

R1: montażowy potencjometr wieloobrotowy 10kΩ
R2, R4, R5, R6, R7: 100kΩ
R3, R8: 10kΩ
R9: R20: 68Ω

Kondensatory

C1..C3: 100nF
C4: 47μF/16V
C5: 150μF/16V
C6: 47μF/16V (niskie ESR, preferowany OS-CON Sanyo)
C7: 4,7μF/16V
C8: 10μF/16V

Półprzewodniki

D1: 1N4148
D2: 1N5817
U3: MAX765
U4: 74HC245
U5: 74HC541
Q1: BC337 itp.
ST1: 7805 (na niewielkim radiatorze z blachy miedzianej)

Różne

LM 24014H: wyświetlacz graficzny z podświetleniem LFOB 08 lub 09
E1449: przetwornica DC/AC do zasilania podświetlenia
gniazdo DB: 25 HM do druku złącze 20: pin (pionowe z łapkami) do druku przycisk miniaturowy do druku zacisk śrubowy ARK: 2 dławik 47μH (tu zastosowany zwykły, zalecany z prądem nasycenia powyżej 0,75A, najlepiej na toroidzie)
akcesoria montażowe: wsporniki wyświetlacza, nóżki, tulejki itd.
przewód połączeniowy 25: żyłowy z wtykami DB: 25 M oraz DB: 25 F.

mógłby być np. domowy sterownik zrobiony ze starej, przeznaczony do skasowania płyty PC - ten temat wielokrotnie przewinał się na liście EP (wtedy raczej posłużymy się DOS-em, ale zasady obsługi wyświetlacza pozostaną te same). Należy wtedy także zadbać o bardziej odporne na zakłócenia dołączenie wyświetlacza (krótsze kable, ekranowana zabudowa, wspólne zasilanie itd.) - w prezentowanym zestawie czasem występują błędy przesyłania (choć nie ma to w tym przypadku poważnego znaczenia).

Jerzy Szczesiul,
jerzy.szczesiul@ep.com.pl