

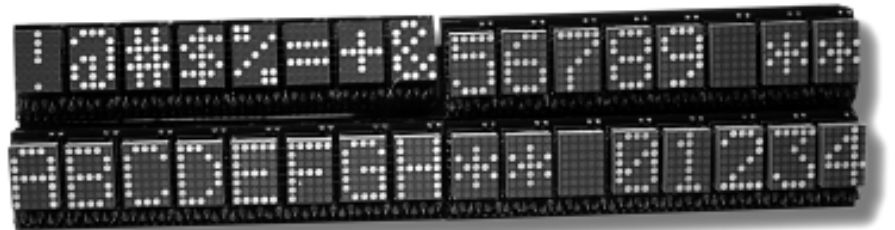
Inteligentny wyświetlacz alfanumeryczny, część 2

kit AVT-324

Kończymy opis modułu inteligentnego wyświetlacza alfanumerycznego.

Opiszemy montaż i uruchomienie układu.

Szczegółowo omawiamy programowanie sterownika oraz możliwość jego zaimplementowania w typowym systemie mikroprocesorowym.



Montaż i uruchomienie

Moduł wyświetlacza składa się z trzech dwustronnych płytek drukowanych. Rozmieszczenie elementów przedstawiono na **rys.9**. Przed rozpoczęciem montażu płytki drukowane należy rozłamać, a następnie wyrównać ich krawędzie przy pomocy pilnika lub papieru ściernego, uważając przy tym, by nie uszkodzić ich powierzchni.

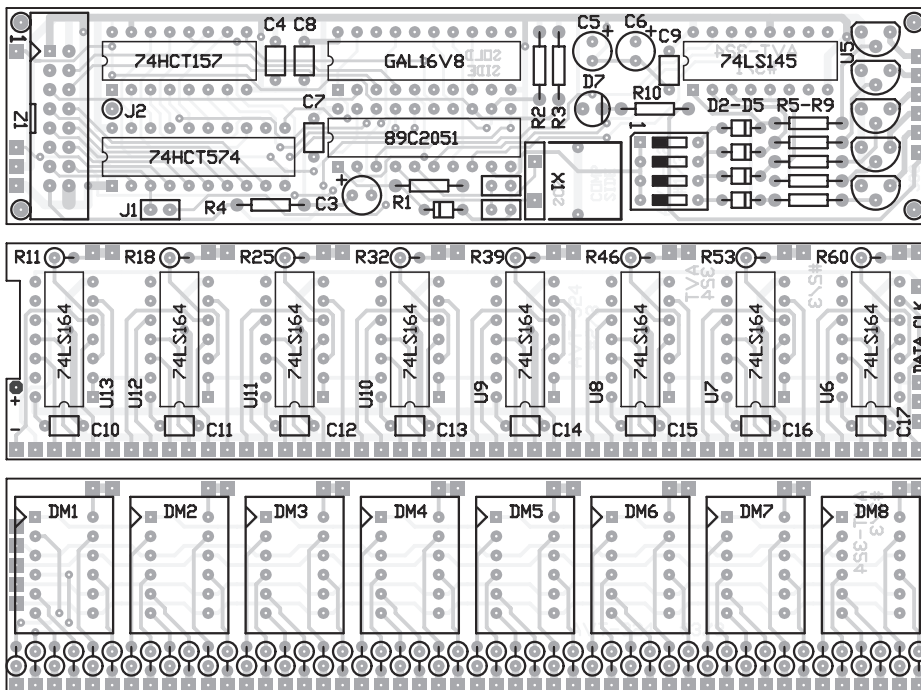
Montaż najlepiej rozpocząć od płytki z matrycami LED. Rezystory R11..R66 (z wyjątkiem rezystorów o wartości 68Ω (z grupy R11..R66) montujemy w pozycji pionowej, zgodnie z obrysem na płytce drukowanej. Po

włożeniu (uwaga na kierunku!) matryc LED, można zabrać się do montażu płytki z rejestrami 74LS164. Pod układy scalone nie należy stosować podstawek. Kondensatory C10..C17 powinny być wlotowane jak najniżej w płytke, tak aby nie wystawały ponad górną powierzchnię obudów układów scalonych. Najlepiej do tego celu nadają się małe kondensatory monolityczne. Często stosowane kondensatory MKT mogą okazać się zbyt wysokie. Pozostałe rezystory 68Ω (z grupy R11..R66) należy wlotować od strony wyprowadzeń płytki, poziomo do jej powierzchni.

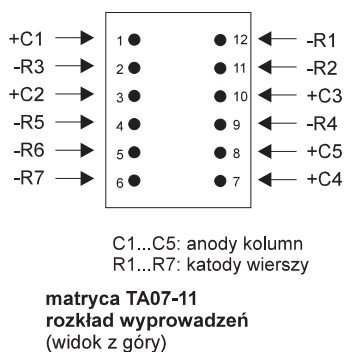
Pozostała jeszcze do montażu płytka z mikroprocesorem U1. Montaż należy rozpocząć od wlotowania elementów niskoprofilowych: rezystorów, diod i podstawek pod układy scalone. Na końcu należy zamontować przełącznik SW1, tranzystory T1..T5 oraz pozostałe elementy. Przy lutowaniu kondensatorów elektrolitycznych należy zwrócić uwagę na ich poprawną polaryzację.

W zależności od potrzeb, należy wlotować rezystor R4. Rezonator kwarcowy X1 montujemy w pozycji leżącej, mocując jego obudowę do płytki przy pomocy dodatkowej klamry, wykonanej z kawałka srebrzanki. Klamrę należy przylutować kroplą cyny do obudowy rezonatora.

Teraz można przystąpić do połączenia wszystkich trzech płytek modułu. Przedtem jednak należy jeszcze raz bardzo



Rys. 9. Rozmieszczenie elementów na płytkach drukowanych.



Rys. 10. Rozmieszczenie wyprowadzeń matrycy TA07-11.

uważnie sprawdzić jakość montażu, eliminując wszelkie możliwe zwarcia oraz przerwy na płytkach drukowanych. Po połączeniu płytek bardzo trudno będzie usunąć pozostawioną niechcący kroplę cyny lub poprawić przerwana ścieżkę na druku.

Do połączenia płytek potrzebne będą typowe, jednorzędowe listwy goldpin. Wszystkie płytki drukowane montujemy w „kanapkę”. Najpierw łączymy płytkę z matrycami LED z płytką rejestrów 74LS164. Tę ostatnią umieszczamy za płytką wyświetlaczy, w pozycji odpowiadającej pokrywaniu się punktów połączeniowych na obrzeżach obu płytek. Jeżeli wszystko przebiegło pomyślnie, można przystąpić do dołączenia płytki sterownika z procesorem U1.

Płytkę tę montujemy „plecami“ do płytki rejestrów, tzn. tak, aby elementy znajdowały się po stronie zewnętrznej (od tyłu) modułu, co ułatwia dostęp do elementów sterownika i umożliwia włożenie układów scalonych U1..U5 w podstawki, po zakończeniu montażu całego modułu.

Do połączenia płytki drukowanej sterownika z pozostałymi płytkami służą punkty lutownicze na brzegach płytki (po lewej i prawej stronie modułu). Niektóre z nich przechodzą poza płytką rejestrów LS164 w specjalnie wyfrezowanej szczeliny. Dotyczy to połączeń kolektorów T1..T5 (patrz schemat z rys.2) z anodami wierszy matryc DM1..DM8. Do tego połączenia należy użyć kawałków srebrzanki, uważając przy tym, aby nie stykały się ze sobą.

Na końcu należy włutować złącze Z1 zgodnie z obrysem na płytce drukowanej sterownika.

Po zmontowaniu całego modułu można przystąpić do jego uruchomienia. Potrzebny będzie zasilacz stabilizowany o napięciu 5V i wydajności prądowej min. 300mA (najlepiej 500mA, np. laboratoryjny). Przed załączeniem zasilania należy umieścić układy scalone w podstawkach na płytce sterownika, a następnie dołączyć punkt testowy oznaczony jako J2 do masy zasilania. Po doprowadzeniu zasilania, np. do złącza Z1 (piny 1 i 2, patrz tabela 1), na wyświetlaczu powinien pojawić się napis: **TEST**. Po krótkiej chwili wykonany zostanie automatyczny test całego wyświetlacza, podczas którego, bez ingerowania w sterowanie, użytkownik ma możliwość skontrolowania działania większości funkcji oferowanych przez moduł. Procedura testująca jest dość złożona, lecz całkowicie bezobsługowa. Nie będziemy jej więc opisywać w artykule.

Jeżeli po włączeniu zasilania wyświetlacz nie zaświeci się, należy skontrolować jeszcze raz jakość wszystkich połączeń. Poprawnie zmontowany układ ze sprawnych elementów działa od razu i nie wymaga dodatkowych czynności uruchomieniowych.

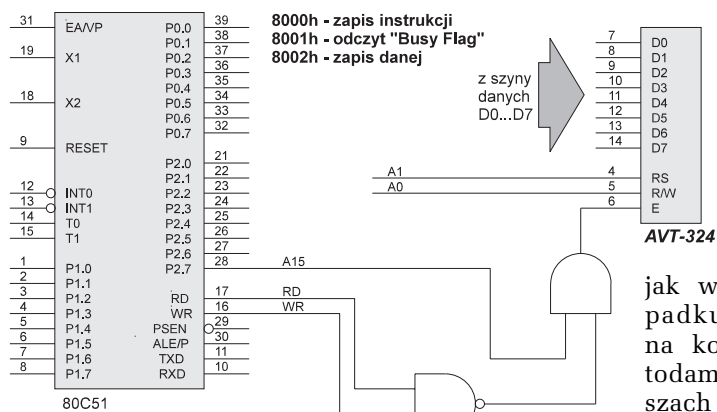
Jeżeli zaś podczas programu testowania zauważymy niepoprawność w wyświetlaniu niektórych znaków lub np. w przypadku zapalenia wszystkich 35 diod matrycy LED nie-

które diody nie będą świeciły, świadczy to o uszkodzonych lub niewłaściwych matrycach DM1..DM8. W tym miejscu przestroga dla osób, które zdecydują się na zakup tych elementów z niepewnego źródła. Autor zaleca stosowanie matryc firmy Kingbright, typ TA07-11. Dalej w oznaczeniu występują trzy litery określające kolor świecenia: „EWA”-

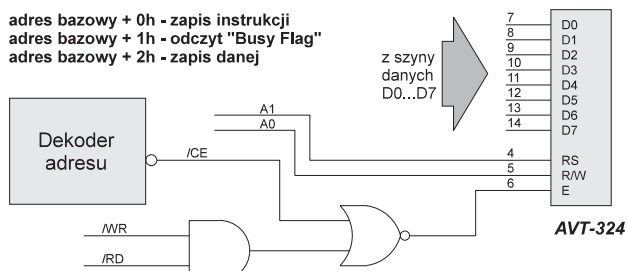
4 bity LSB \ 4 bity MSB	0000	0001	0010	0011	0100	0101	0110	0111
xxxx 0000	(1)	A	B	C	D	E	F	G
xxxx 0001	(2)	0	1	2	3	4	5	6
xxxx 0010	(3)	7	8	9	*	*	*	*
xxxx 0011	(4)	*	*	*	*	*	*	*
xxxx 0100	(5)	*	*	*	*	*	*	*
xxxx 0101	(6)	*	*	*	*	*	*	*
xxxx 0110	(7)	*	*	*	*	*	*	*
xxxx 0111	(8)	*	*	*	*	*	*	*
xxxx 1000	(1)	*	*	*	*	*	*	*
xxxx 1001	(2)	*	*	*	*	*	*	*
xxxx 1010	(3)	*	*	*	*	*	*	*
xxxx 1011	(4)	*	*	*	*	*	*	*
xxxx 1100	(5)	*	*	*	*	*	*	*
xxxx 1101	(6)	*	*	*	*	*	*	*
xxxx 1110	(7)	*	*	*	*	*	*	*
xxxx 1111	(8)	*	*	*	*	*	*	*

UWAGA:
 (1)..(8) - znaki definiowane przez użytkownika.
 Bit 7 kodu znaku z tabeli odpowiada za atrybut migania.

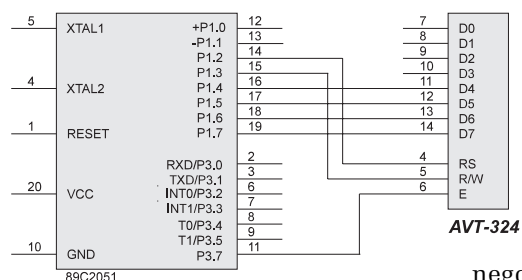
Rys. 11. Zawartość generatora znaków.



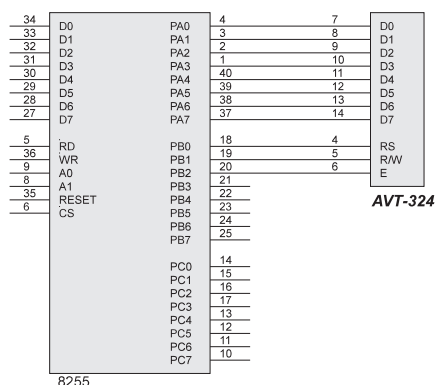
Rys. 12a. Sposób podłączenia modułu do procesora '51.



Rys. 12b. Sposób podłączenia modułu do dowolnego systemu μ P.



Rys. 12c. Sposób podłączenia modułu do procesora 89C2051.



Rys. 12d. Sposób podłączenia modułu do portów 8255.

czerwony, „GWA“-zielony lub „YWA“- żółty“.

Przy zakupie matryc innych producentów należy upewnić się, czy rozkład wyprowadzeń był taki sam, jak dla zalecanych w artykule elementów. Sprawa szczegól-

nie dotyczy właściwej polaryzacji kolumn i wierszy matrycy. W handlu spotykane są dwa rodzaje matryc: takie

jak w naszym przypadku, z anodami na kolumnach i katodami LED w wierszach (patrz rys.4), oraz drugi rodzaj, którego polaryzacja jest odwrotna. Strzeżmy się zatem nieuczciwych sprzedawców i sprawdźmy przed zakupem, np. za pomocą baterijki, czy nabywamy właściwe elementy. Odpowiednie podłączenie zasilania w celu sprawdzenia matrycy ułatwi nam rozkład wyprowadzeń pokazany na rys.10.

Jeżeli procedura testowa wypadła pomyślnie, to połączenie J2 należy usunąć, a następnie ustawić za pomocą SW1 właściwy adres (offset) naszego modułu. Jeżeli korzystamy tylko z jednego urządzenia, wszystkie przełączniki SW1 powinny pozostać w pozycji OFF (rozwarne). W przeciwnym przypadku w kolejnych modułach ustawiamy właściwy im adres. Nie zapomnijmy także o odpowiednim umieszczeniu jumpera w zworze J1 (patrz opis układu i rys.2).

Kompatybilność układu AVT-324 ze sterownikiem LCD HD44780 umożliwia bezpośrednie dołączenie urządzenia w miejsce wyświetlacza ciekłokrystalicznego.

Ze względu na spory pobór prądu przez nasz moduł, należy sprawdzić możliwości dotychczas używanego zasilacza.

Kabel połączeniowy wykonujemy przy użyciu typowych złączy AFC-16 oraz odcinka taśmy 16-żyłowej. Od strony układu sterowania wtyk może być typu AFC-14, bowiem jak opisano wcześniej, dwie końcówki na złączu Z1 (15 i 16) nie są częścią interfejsu wejściowego.

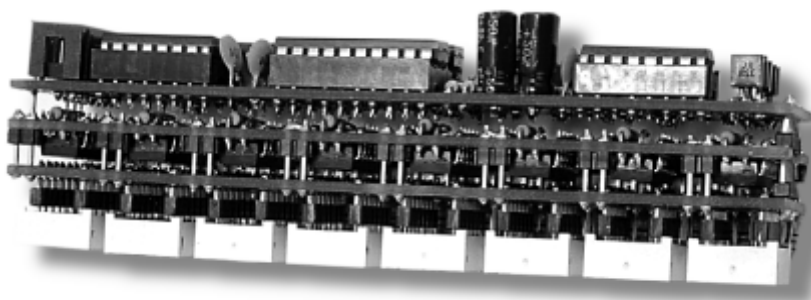
W wypadku łączenia kilku modułów równolegle, na jednym dłuższym odcinku taśmy montujemy potrzebną ilość złączy AFC-16, w odległościach umożliwiających podłączenie modułów w wymaganej konfiguracji. Na zdjęciu widać układ w wersji 2 linie po 16 znaków (4 moduły AVT-324).

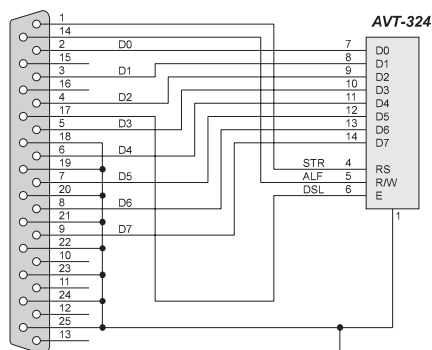
Programowanie i obsługa

W tej części artykułu zajmiemy się sposobem programowania modułów AVT-324 oraz podamy kilka sposobów na dołączanie ich do różnych systemów mikroprocesorowych.

Na początku zapoznajmy się z zestawem znaków alfanumerycznych wbudowanym w pamięć generatora znaków CG ROM. Tabela przedstawiona na rys.11 pokazuje wszystkie 112 predefiniowanych znaków wraz z ich kodami (zgodnymi z ASCII). Polskie litery zostały zapisane pod kodami: 10h..1Fh (16..31 dziesiętnie). Kody znaków 00h..0Fh są puste i służą do generowania dodatkowych znaków użytkownika. Możliwe jest zdefiniowanie maksymalnie 8 takich znaków. Sposób ich generowania opiszemy w dalszej części artykułu.

Tabela kończy się na kodzie 127 (7Fh). W oryginalnym generatorze znaków sterownika LCD -





PORT DRUKARKOWY PC (Centronics)

Rys. 13. Podłączenie modułu wyświetlacza do portu Centronics.

HD44870, kody powyżej 127 zawierają znaki z alfabetu japońskiego, co z naszego punktu widzenia jest nieprzydatne. Dlatego w naszym urządzeniu podanie kodu wyższego niż 127 (128..255) powoduje wyświetlenie znaku jak z tabeli (z wyzerowanym najstarszym bitem) lecz z atrybutem migotania. Ta dodatkowa funkcja nie występuje we wspomnianych sterownikach Hitachi, lecz w naszym urządzeniu może być z korzyścią wprowadzone.

Dla przykładu, wpisanie znaku o kodzie AFh (175 dziesiętnie) spowoduje wyświetlenie migoczącej litery „G“, bo przecież: AFh - 80h = 2Fh - kod litery „G“ (175-128 = 47), itd.

Na rys.12 pokazano kilka przykładowych sposobów na dołączenie modułu AVT-324 do zewnętrznych urządzeń sterujących.

Pierwszy przykład (z rys.12a) bazuje na systemie mikroprocesorowym opartym o popularny kontroler 80C51, pracujący z zewnętrzną pamięcią danych. W tym przypadku nasz moduł jest wybierany począwszy od adresu 8000h (linia adresowa A15=1). Adresy sterowania podane są na rysunku.

W drugim przypadku (z rys. 12b) mamy do czynienia z podobnym systemem, lecz jako sygnał selekcji wykorzystujemy sygnał wyboru /CE z dekodera adresowego. Polaryzacja sygnału wyboru jest w tym przypadku ujemna. Przeporządkowanie poszczególnych operacji do adresów jest podobne jak poprzednio, z tym, że adres obliczany jest jak pokazano na rysunku. Adres bazowy jest generowany bezpośrednio przez dekodery.

Przykład z rys.12c ilustruje sterowanie wyświetlaczem bezpośrednio za pośrednictwem portów mikroprocesora. Wykorzystywany jest tryb transmisji danej 4-bitowej. Procesor najpierw ustawia odpowiednią daną oraz sygnały sterujące R/W i RS, a następnie podaje dodatni impuls na końcówkę P3.7 (11), co w efekcieysterowuje wyświetlacz.

W ostatnim przykładzie (z rys.12d) pokazano sterowanie wyświetlaczem przy pomocy sygnałów z uniwersalnego portu typu 8255. Port A steruje zapisem danej do wyświetlacza oraz odczytuje stan jego zajętości (Busy Flag na linii PA7). Trzy spośród 8 sygnałów portu PB bezpośrednio sterują wejściami R/W, RS i E.

Amatorzy programowania na

Tabela 2. Zestawienie poleceń wyświetlacza AVT-324.

Instrukcja	Dane										Opis	Czas wykonania
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Czyści wyświetlacz i ustawia kursor na początku (adres=0)	<200µs
Return home	0	0	0	0	0	0	0	0	1	*	Ustawia kursor na pozycji początkowej (adres=0). Jeżeli napis był przesunięty, ustawia go na pozycji oryginalnej. Zawartość DD RAM nie ulega zmianie.	<200µs
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Ustawia kierunek przesuwu kursora i włącza/wyłącza funkcję przesuwu napisu przy zapisie do DD RAM.	<200µs
Display ON/OFF	0	0	0	0	0	0	1	D	C	B	Włącza/wyłącza wyświetlacz (D), kursor (C) i funkcję migania kursora (B) - "blink".	<200µs
Cursor & display shift	0	0	0	0	0	1	S/C	R/L	*	*	Przesuwa kursor i napis bez zmiany zawartości DD RAM.	<200µs
Function set	0	0	0	0	1	DL	N	*	*	*	Ustala ilość bitów danych interfejsu (DL) oraz liczbę wyświetlanych linii (L)	<200µs
Set CG RAM address	0	0	0	1	A _{CG}						Ustawia adres w CG RAM. Wszystkie operacje zapisu danej odnoszą się do tej pamięci po wywołaniu tej instrukcji.	<200µs
Set DD RAM address	0	0	1	A _{DD}						Ustawia adres w DD RAM. Wszystkie operacje zapisu danej odnoszą się do tej pamięci po wywołaniu tej instrukcji	<200µs	
Read busy flag	0	1	BF	*	*	*	*	*	*	*	Odczytuje stan flagi zajętości "Busy Flag".	<1µs
Write data to CG or DD RAM	1	0	Dana do zapisu								Zapisuje daną (znak) do pamięci DD RAM lub CG RAM.	<200µs

I/D = 1: Zwiększenie (+1),
 I/D = 0: Zmniejszenie (-1),
 S = 1: Towarzyszy przesuwaniu napisu,
 S/C = 1: Przesuwanie napisu,
 S/C = 0: Przesuwanie kursora,
 R/L = 1: Przesuwanie w prawo,
 R/L = 0: Przesuwanie w lewo,
 DL = 1: Interfejs 8-bitowy,
 DL = 0: Interfejs 4-bitowy,
 N = 1: 2 linie N = 0: 1 linia,
 BF = 1: Układ zajęty,
 BF = 0: Gotowy na następną instrukcję,

DD RAM: Pamięć znaków (wyświetlania)
 CG RAM: Pamięć generatora znaków
 A_{CG}: Adres w pamięci CG RAM
 A_{DD}: Adres w pamięci DD RAM

* - gwiazdką oznaczono bity nie znaczące

Typ pamięci	Adres	Przeznaczenie
Pamięć CG ROM	00h..07h	kody 8 znaków użytkownika, powtórzone w obu zakresach
	08h..0Fh	polские litery duże i małe
	10h..1Fh	pozostałe znaki ASCII
	20h..7Fh	jak dla zakresu 00h..7Fh lecz z atrybutem migotania znaku
Pamięć CG RAM	00h..3Fh	pamięć matryc znaków zdefiniowanych przez użytkownika: 64 bajty, po 8 na każdy znak, patrz tabela 6.
Pamięć DD RAM	00h..27h	1-liniowy tryb pracy, maks. 40 znaków
	00h..27h 40h..67h	2-liniowy tryb pracy, maks. 80 znaków

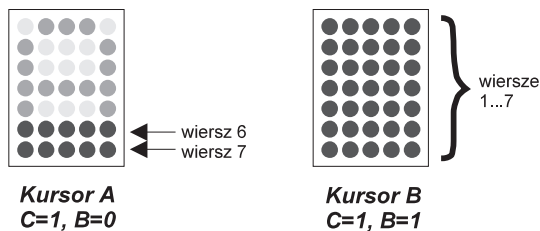
PC do sterowania naszymi modułami mogą wykorzystać łącze drukarkowe standardu Centronics. Najprostszy sposób dołączenia wyświetlacza przedstawia rys.13. Przykładowa procedura w języku Turbo Pascal realizująca zapis znaku do wyświetlacza z wykorzystaniem przedstawionego połączenia może być następująca:

```
port[LPT+$02]:=$08;
{ ustawienie sygnału E=0 }
port[LPT+$02]:=$09;
{ ustawienie RS=1, R/W=0 }
port[LPT+$00]:=kod_znaku;
{ zapis znaku na linii D0..D7}
port[LPT+$02]:=$01; { E=1 }
port[LPT+$02]:=$08;
{ E=0 koniec zapisu znaku }
delay(1); { poczekać na wykonanie instrukcji }
gdzie LPT: adres bazowy portu drukarkowego (= 378h dla LPT1, = 278h dla LPT2).
```

W tym jednak przypadku nie jest możliwe monitorowanie flagi zajętości Busy Flag. Dlatego po wykonaniu jakiegokolwiek operacji należy odczekać np. 1 ms - instrukcja „delay(1)“.

Zestawienie wszystkich poleceń wyświetlacza AVT-324 znajduje się w tabeli 2.

Natomiast tabela 3 zawiera definicje obszarów adresowych pamięci generatora znaków CG ROM, generatora znaków użytkownika



Uwaga: oba typy kursorów migoczą!
Rys. 14. Przykładowe kształty kursora.

CG RAM oraz pamięć wyświetlania DD RAM. Poniżej opisujemy działanie poszczególnych instrukcji. Pozycja bitów oznaczonych gwiazdką nie ma znaczenia.

„Clear display“
- czyszczenie

wyświetlacza
RS R/W D7..D0
0 0 0 0 0 0 0 0 0 1
Cała pamięć DD RAM zostaje wypełniona spacjami (20h), wskaźnik (kursor) adresu DD RAM zostaje wyzerowany. Jeżeli napis był „przesunięty“, wraca na swoje miejsce. Rozkaz ustawia bit I/D w słowie „Entry Mode“. Bit „S“ w tym słowie się nie zmienia.

„Return home“
- ustawienie kursora na poz. początkowej
RS R/W D7..D0
0 0 0 0 0 0 0 0 1 *
Zeruje wskaźnik adresu DD RAM, kursor zostaje przesunięty do pozycji 0. Przesunięty tekst powraca na swoje miejsce, zawartość pamięci DD RAM nie ulega zmianie.

„Entry mode set“
- sposób sterowania wyświetlaczem
RS R/W D7..D0
0 0 0 0 0 0 0 1 I/D S
I/D: inkrementuje (I/D=1) lub dekrementuje (I/D=0) wskaźnik adresu DD RAM o 1 po każdorazowym zapisie znaku do tej pamięci. Kursor zostaje przesunięty w prawo, gdy I/D=1, lub w lewo, gdy I/D=0. To samo dotyczy pamięci CG RAM przy zapisie matrycy znaku użytkownika.

S: powoduje przesuwanie całej zawartości DD RAM (napisu) w lewo lub prawo, w zależności od bitu I/D. W praktyce wygląda to tak, jakby kursor stał w miejscu, a cały napis się przesuwał. Zapis do CG RAM przy S=1 nie powoduje przesuwania się napisu.

S/C	R/L	Efekt
0	0	Przesuwa kursor w lewo, wskaźnik adresu zostaje zmniejszony o 1
0	1	Przesuwa kursor w prawo, wskaźnik adresu zostaje zwiększony o 1
1	0	Przesuwa cały napis w lewo, kursor podąża za przesuwany tekst
1	1	Przesuwa cały napis w prawo, kursor podąża za przesuwany tekst

„Display ON/OFF“
- włączanie wyświetlacza i kursora
RS R/W D7..D0
0 0 0 0 0 0 1 D C B
D: włącza wyświetlacz, gdy D=1, i gasi, gdy D=0. Zmiana tego bitu nie powoduje zmiany zawartości DD RAM (wprowadzonego tekstu).

C: pokazuje kursor, gdy C=1, i chowa, gdy C=0. Nawet gdy kursor zostanie ukryty, jest nadal aktywny i podąża wraz ze wskaźnikiem pamięci DD RAM podczas operacji zapisu. Kursor „zajmuje“ dwie dolne linie matrycy LED jak to pokazano na rys.14.

B: po ustawieniu tego bitu (B=1) kursor zajmuje całą matrycę, dodatkowo migocząc na przemian ze znajdującym się „za nim“, wyświetlanym znakiem.

Częstotliwość migotania kursora jest taka sama jak znaku i wynosi ok. 2Hz.

„Cursor & display shift“
- kontrola kursora i przesuwania tekstu

RS R/W D7..D0
0 0 0 0 0 1 S/C R/L * *
Przesuwa kursor lub napis w prawo lub lewo, bez zmiany zawartości pamięci DD RAM. W 2-liniowym trybie pracy kursor przechodzi do drugiej linii w momencie minięcia 40 pozycji w pierwszej linii (adres 27h). W tym trybie, przy przesuwaniu napisu, obie linie są przesuwane równolegle i niezależnie. Oznacza to, że ostatni znak w 1 linii trafia na miejsce pierwsze w tej samej linii, a nie przechodzi do linii drugiej. To samo dotyczy linii nr 2. W praktyce wygląda to jak przesuwanie poziome dwóch niezależnych napisów w 2 liniach. W tabeli 4 zestawiono działanie kombinacji bitów S/C i R/L.

Tabela 5. Kolejne instrukcje przy zapisie przykładowego komunikatu.

No	Instrukcja, RS, R/W D7..D0	Wyświetlacz	Operacja
1	włączenie zasilania (inicjalizacja obwodów wewnętrznych)		wyświetlacz wygaszony
2	"Function set", 0,0 00110000		Ustawienie interfejsu 8-bit, tryb 1-linia
3	"Display ON/OFF", 0,0 00001110	-	Włączenie wyświetlacza i zapalenie kursora
4	"Entry mode set", 0,0 00000110	-	Tryb inkrementacji wskaźnika adresu, bez przesuwania całego tekstu
5	"Write data to CG/DD RAM", 1,0 00100100	\$ _	wpisanie znaku "\$"
6	"Write data ...", 1,0 01010011	\$S _	wpisanie znaku "S"
7	"Write data...", 1,0 00010011	\$SŁ _	wpisanie polskiego znaku "Ł"
8	"Write data...", 1,0 01000001	\$SŁA _	wpisanie znaku "A"
9	"Write data...", 1,0 01010111	\$SŁAW _	wpisanie znaku "W"
10	"Write data...", 1,0 01000101	\$SŁAWE _	wpisanie znaku "E"
11	"Write data...", 1,0 01001011	\$SŁAWEK _	wpisanie znaku "K"
12	"Write data...", 1,0 00100100	\$SŁAWEK\$	wpisanie znaku "\$"
13	"Return home", 0,0 00000010	\$SŁAWEK\$	powrót kursora na pozycję początkową, adres "0"

„Function Set“

- ustawienie funkcji dodatkowych
 RS R/W D7..D0
 0 0 0 0 1 DL N * * *

DL: ustala szerokość magistrali danych. Gdy DL=1, dane przesyłane są w postaci 8-bitowej linii D0..D7. Kiedy DL=0, transmisja jest 4-bitowa: linie D4..D7. Gdy wybrany jest interfejs 4-bitowy, każda dana lub rozkaz musi być przesłana w 2 cyklach, najpierw starsza część bajtu, potem młodsza. Po każdej operacji należy sprawdzić „Busy Flag“ lub odczekać czas określony w tabeli 1.

N: ustala tryb pracy 1-liniowy (N=0), lub 2-liniowy (N=1). Gdy aktywny jest tryb 1-liniowy, a niektóre z modułów mają fizycznie (SW1) ustawiony adres drugiej linii, to pozostają one nie używane.

„Set CG RAM Address“

- ustawienie adresu pamięci znaków użytkownika

RS R/W D7..D0
 0 0 0 1 A A A A A A

Ustala adres aktualnego zapisu do pamięci matrycy znaku użytkownika CG RAM.

Dozwolony adres: 00h..3Fh jak podano w tabeli 3. Po tej operacji

dane będą umieszczane od ustalonego adresu w CG RAM. (AAAAAA - 6-bitowy adres)

„Set DD RAM Address“

- ustawienie adresu pamięci tekstu

RS R/W D7..D0
 0 0 1 A A A A A A A

Ustala adres aktualnego zapisu do pamięci tekstu DD RAM. Po tej operacji dane są umieszczane od ustalonego adresu w DD RAM. (AAAAAAA-7-bitowy adres)

Gdy N=0 (tryb 1-liniowy) dozwolony zakres adresu: 00h..27h, gdy N=1 (tryb 2-liniowy) adresy 1 linii: 00h..27h, 2 linii: 40h..67h

„Read busy flag“

- odczyt flagi zajętości
 RS R/W D7..D0
 0 1 BF * * * * * *

Odczytanie stanu flagi zajętości „Busy Flag“. Gdy po odczycie BF=1 znaczy to, że moduł wykonuje wewnętrzną operację i nie przyjmie danej ani instrukcji. Następne dane powinny być przesyłane do wyświetlacza gdy BF=0.

„Write data to CG or DD RAM“

- zapis danej do CG RAM lub DD

RAM
 RS R/W D7..D0
 1 0 D D D D D D D D

Wpisuje 8-bitową daną DDDDDDDD do pamięci tekstu DD RAM lub generatora znaków użytkownika CG RAM. To, do jakiej pamięci zostaje zapisana dana, zależy od tego, do jakiej pamięci odnosiło się ostatnie ustawienie adresu, patrz instrukcje „Set CG RAM address“ i „Set DD RAM address“. Po zapisie do pamięci DD RAM lub CG RAM wskaźnik adresu zostaje automatycznie inkrementowany lub dekrementowany o 1 w zależności od ustawienia polecenia „Entry Mode“.

Przykładowa kolejność instrukcji w celu zapisania 8-znakowego tekstu np.: „\$SŁAWEK\$“ jest pokazana w tabeli 5.

Inicjalizacja modułu po włączeniu zasilania trwa około 20 ms. Czas ten nie uwzględnia okresu narastania napięcia zasilającego do wartości 4,6V. Czas ten jest potrzebny na ustawienie rejestrów konfiguracyjnych mikroprocesora U1. Potem moduł jest gotowy do przyjęcia instrukcji lub danej. Wartości początkowe bitów sterujących wyświetlaniem są następujące:

- a) wyświetlacz jest wygaszony
- b) „Function Set“: DL=1 (8-bit), N=0 (1-linia);
- c) „Display ON/OFF“: D=0, C=0, B=0;
- d) „Entry Mode“: I/D=1 (+1), S=0 (bez przesuwania).

Wewnętrzna pamięć tekstu DD RAM jest wypełniona spacjami (znakami pustymi), natomiast pamięć matryc znaków użytkownika CG RAM wypełniona jest przypadkowymi danymi.

Generowanie znaków

Jak przewiduje standard sterowników LCD, nasz moduł posiada 64 bajty pamięci CG RAM, czyli pamięci generatora znaków użytkownika. Zapis do tej pamięci powoduje generowanie, zgodnie z tabelą 1, kolejnych bajtów wzoru znaków, począwszy od wiersza nr 1 matrycy, a na siódmym skończywszy.

Powiązanie między adresami w CG RAM, DD RAM, a matrycą znaku przedstawiono w tabeli 6.

Tabela 6.

Kod znaku (tab.1) bity: 7 6 5 4 3 2 1 0	Adres w CG RAM bity: 5 4 3 2 1 0	Matryca znaku bity: 7 6 5 4 3 2 1 0		
0000*000 kod: 00h lub 08h	000	000	***	01010
		001	***	10101
		010	***	01010
		011	***	10101
		100	***	01010
		101	***	10101
		110	***	01010
		111	***	00000
0000*001 kod: 01h lub 09h	001	000	***	00000
		001	***	00100
		010	***	01110
		011	***	11111
		100	***	01110
		101	***	00100
		110	***	00000
		111	***	00000
itd.... aż do kodu 07h				
0000*111 kod: 07h lub 0Fh	111	000	***	11111
		001	***	00000
		010	***	11111
		011	***	00000
		100	***	11111
		101	***	00000
		110	***	11111
		111	***	00000

Dla przykładu, jako pierwszy znak (kod=00h) zdefiniowano „szachownicę“, a pod kodem 02h znak „karo“, zaś ostatnim zdefiniowanym znakiem są cztery poziome linie. Bity oznaczone gwiazdką nie mają znaczenia przy definiowaniu matrycy znaku.

Ósmy wprowadzany kolejno bajt (wiersz) matrycy znaku nie jest istotny. Ze względu na zastosowane matryce LED: 5x7, zwykle należy wpisać 00h.

Kompatybilność z HD44780

Poniżej przedstawiono różnice w programowaniu i obsłudze przedstawionego modułu wyświetlacza ze sterownikami HD44780 stosowanymi w tekstowych wyświetlaczach LCD.

1. W rozkazie „Function Set“ nie jest implementowany

bit 2, nazywany często „F“. Bit ten określa rodzaj matrycy znaku do wyświetlenia. W naszym module zastosowanie matryc LED 5x7 z oczywistych względów wykluczyło użycie tej opcji.

2. Kursor w module AVT-324 może pojawiać się w dwóch wariantach, jak przedstawiono na rys.14. W każdym przypadku kursor jest w trybie migotania. Dlatego ustawienie np. C=0, B=1 (kursor wyłączony, migotanie włączone) nie powoduje migotania znaku na pozycji kursora. Do tego celu użyto kodów ASCII z rys.11, gdzie, aby uzyskać efekt migotania znaku, należy ustawić najstarszy bit kodu podstawowego.

3. W wyświetlaczach LCD przy odczycie stanu „Busy Flag“ można także sprawdzić stan licznika adresu wyświetlacza „AC-address counter“. U nas, ze względów na przyjęte uproszczenia interfejsu wejściowego, funkcja ta nie jest dostępna. W praktyce jednak nie jest to dużą przeszkodą, bowiem zazwyczaj pozycja wskaźnika adresu jest z reguły pamiętana w programie sterownika obsługującego wyświetlacz.

4. Podobnie jak w pkt.3, nie ma możliwości odczytu danej („Read Data“) z pamięci CG RAM lub DD RAM,

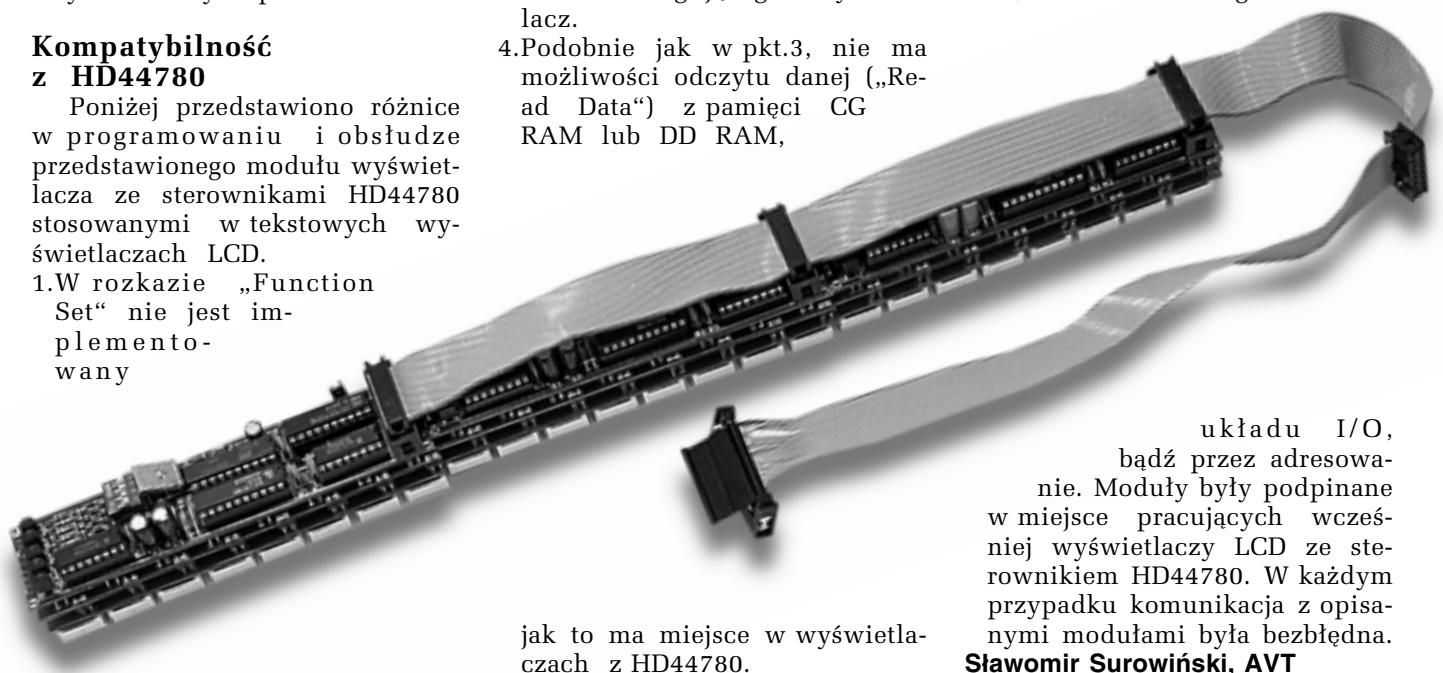
Operacja taka jest dostępna w LCD przy kombinacji sygnałów: RS=1, R/W=1.

5. Przy sterowaniu modułu AVT-324 w trybie transmisji 4-bitowej (DL=0), po wpisaniu każdej połówki bajtu danej lub rozkazu, należy bezwzględnie sprawdzić stan „BUSY FLAG“ lub odczekać okres czasu zgodnie z tabelą 2. W pierwowzorze LCD obie połówki bajtu można wpisywać bez kontroli flagi zajętości.

6. Ograniczenie obszaru pamięci przy sterowaniu wyświetlacza 1-liniowego zostały wyjaśnione w poprzedniej części artykułu, należy o tym jednak pamiętać przy tworzeniu oprogramowania sterującego.

7. W modułach LCD fizyczne zatrzaśnięcie danej z linii D0..D7 następuje przy opadającym zboczach sygnału E (EN). W naszym urządzeniu zapis następuje przy zboczu narastającym. Nie jest to przeszkodą w kompatybilności od strony sterowania, bowiem przebiegi czasowe pokazane na rys.6 są mniej krytyczne dla modułu AVT-324.

8. W laboratorium autor przetestował działanie wyświetlacza zmontowanego z 4 modułów AVT-324 w konfiguracji 1x32 oraz 2x16 znaków. Urządzenia były sterowane czterema sposobami: przez bezpośrednie sterowanie z portów procesora, uniwersalnego



jak to ma miejsce w wyświetlaczach z HD44780.

układu I/O, bądź przez adresowanie. Moduły były podpinane w miejsce pracujących wcześniej wyświetlaczy LCD ze sterownikiem HD44780. W każdym przypadku komunikacja z opisanymi modułami była bezbłędna.

Sławomir Surowiński, AVT