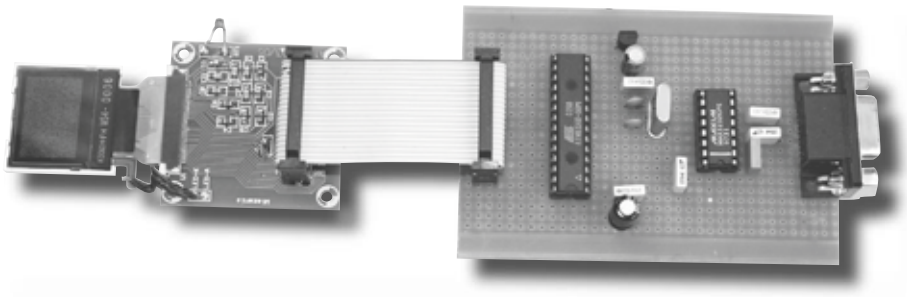
Obsługa kolorowego wyświetlacza graficznego 96x64, część 2

Po upowszechnieniu się wyświetlaczy LCD monochromatycznych, nadchodzi era wyświetlaczy kolorowych. W artykule przedstawiamy sposób wykorzystania małego wyświetlacza kolorowego LCD z liczbą kolorów wynoszącą 65k!

Rekomendacje:

przejście z wyświetlaczy monochromatycznych na kolorowe, z całą pewnością zwiększy atrakcyjność projektowanych urządzeń, a w niektórych wypadkach podniesie ich walory użytkowe. Warto więc zapoznać się ze sposobem ich sterowania, który - jak pokazuje autor - wcale nie jest trudny.



Konfiguracja wyświetlacza

Na przykładzie programu z list. 1 zostaną przedstawione instrukcje konfiguracyjne potrzebne do wyświetlenia, w tym wypadku, kolorowej bitmapy.

Po włączeniu zasilania następuje wyzerowanie wyświetlacza LCD poprzez podanie na jego linię zerującą stanu niskiego na czas 20ms.

Następnie do wyświetlacza jest wysyłana instrukcja włączenia oscylatora o adresie R00h i wartości 0001h. W tab. 3 przedstawiono wygląd rejestru adresowego i jak widać do adresowania (wyboru rejestrów) służy tylko 7 bitów. Po włączeniu oscylatora należy odczekać czas ponad 10ms (w programie dla bezpieczeństwa 20ms) potrzebny na jego ustabilizowanie się.

Kolejne dwie instrukcje wysyłają dane 128Ch (00010010 10001100 bin) do rejestru R03h i 0000h do rejestru R0Ch, które są odpowiedzialne za kontrolę napięć sterujących wyświetlaczem. W tab. 3 widoczne są bity tych dwóch rejestrów.

Bity BS2-0 określają wypełnienie napięcia polaryzacji. Dla użytego wyświetlacza wartość ta powinna wynosić 1/9 (wartości bitów 010).

Bity BT1-0 konfigurują wartości napięć z przetwornic. Wartości te zostały skonfigurowane dla przetwornicy 1 jako Vci1x2 i dla przetwornicy 2 jako Vci2x2,5.

Bity DC2-0 są odpowiedzialne za częstotliwość obsługi przetwornic wyświetlacza. Wartości tych bitów zostały ustawione na 100, czyli przetwornice będą obsługiwane z podziałem częstotliwości przez 16 i 32.

Bity AP2-0 określają wartość pobieranego prądu przez wewnętrzne przetwornice. Wartości bitów ustawione zostały na 011, czyli pobierany prąd dla obu przetwornic będzie wynosił 100%.

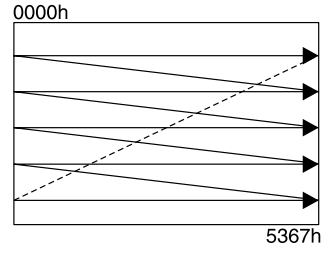
Bit SLP został wyzerowany. Ustawienie tego bitu wprowadza wyświetlacz w tryb *Sleep* (oscylator pracuje dalej).

Bit STB także został wyzerowany. Ustawienie tego bitu wprowadza wyświetlacz w tryb *Standby* (oscylator zostaje zatrzymany). Aby wyjść z tego trybu należy wyzerować wyświetlacz.

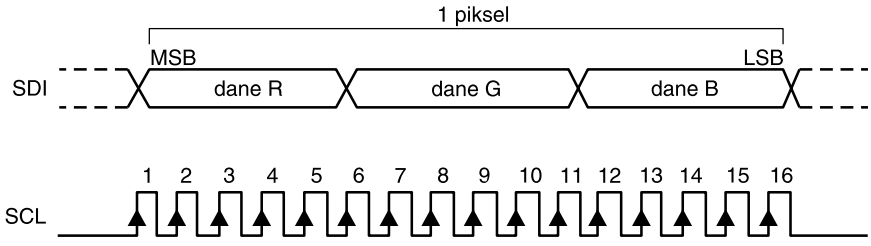
Bity VC2-0 określają wartość napięcia Vci1. Bity zostały wyzerowane, więc wartość napięcia Vci1 będzie wynosić 1,00xVciL. Zalecane wartości przedstawionych ustawień zazwyczaj są podawane w karcie katalogowej wyświetlacza.

Następnie pod adres R04h (rejestr kontrastu – tab. 3) wysyłana zostaje wartość 1AE0h (00011010 11100000 bin).

Bit VRCNT określa czy napięcie odniesienia VREFOUT będzie łączone bezpośrednio do VREFLCD, czy poprzez zewnętrzny obwód kompensacji temperaturowej. Ponieważ wykorzystana została opcja kompensacji temperaturowej poprzez zewnętrzny termistor, więc bit ten został wyzerowany.



Rys. 6. Sposób zapisu danych do pamięci GRAM



Rys. 7. Przesyłanie danych o kolorach

Bity VR4-0 określają napięcie odniesienia VREFH. Bity te zostały ustawione na wartość 11010, czyli napięcie VREFH będzie wynosić VREFLCDx7,5.

Bit VRON określa źródło napięcia odniesienia. Bit ten został ustawiony, co oznacza wybranie wewnętrznego napięcia odniesienia.

Bity CT6-0 określają kontrast, który można regulować w 128 krokach. W przykładzie wartość kontrastu została ustalona na 1100000.

Dalej, do rejestru R01h, który jest rejestrem konfigurującym pracę bufora sterującego matrycą wyświetlacza (tab.3) wysyłana jest wartość 0207h (00000010 00000111bin).

Bity CSFT i CMS umożliwiają wybór formy skanowania (multipleksowania) matrycy ekranu. W przykładzie CSFT równy jest 0, a CMS równy 1, więc jednocześnie będą odświeżane linie parzyste i nieparzyste wyświetlacza począwszy od jego dolnej części ekranu. Można także tymi bitami określić inny kierunek skanowania oraz wybrać taki tryb, by najpierw były odświeżane tylko linie parzyste, a następnie tylko nieparzyste.

Bit SGS określa kierunek wybierania segmentów wyświetlacza. W przykładzie bit ten został wyzerowany, więc segmenty będą wybierane od SEG1 do SEG288.

Bit 4L określa formę prezentacji na wyświetlaczu linii (wiersza). Ustawienie tego bitu powoduje, że jedna linia jest pokazywana jako 4 linie. W przykładzie bit ten został wyzerowany.

Bity NL4-0 określają współczynnik wypełnienia sygnału sterującego matrycą wyświetlacza. Dla wyświetlacza o 64 wierszach wypełnienie powinno wynosić 1/64, czyli wartości tych bitów powinny zostać ustawione na 00111.

Następnie do rejestru kontrolnego przebiegów sterujących ekranem o adresie R02h (tab.3) zostaje zapisana wartość 0000h. Bity tego rejestru są odpowiedzialne za specyficzne sterowanie matrycy LCD, które daje polepszenie obrazu przy większych rozdzielczościach ekranu. Wszystkie bity tego rejestru zostały wyzerowane, co jest jednoznaczne z brakiem wykorzystania tych funkcji, które mogą się przydać przy większych rozdzielczościach wyświetlacza. Przy wyzerowanym bicie B/C pozostałe bity są nieaktywne.

Dalej, do rejestru trybu pracy wyświetlacza o adresie R05h wysłana zostaje wartość 0210h (00000010 00010000bin), a do rejestru porównania o adresie R06h zostaje wysłana wartość 0000h. Oba rejestry zostały przedstawione w tab.3.

Bity SPR1-0 określają liczbę kolorów wyświetlacza. W przykładzie oba bity zostały wyzerowane, przez co dostępnych będzie 65000 kolorów. Można także wybrać liczbę kolorów równą 4096 lub 256.

Kiedy bit HWM jest ustawiony, wtedy dane do pamięci obrazu GRAM mogą być przesyłane w trybie szybkim (jednocześnie przesyłane są 4 słowa), który jest dostępny tylko przy pracy wyświetlacza z interfejsem równoległym. W tym przypadku wartość tego bitu jest bez znaczenia.

Bity I/D1-0 określają czy adres pamięci GRAM (pamięci obrazu) będzie automatycznie inkrementowany czy dekrementowany.

Natomiast bit AM określa sposób zapisywania danych w pamięci GRAM. Przy AM=0 dane będą zapisywane poziomo, a przy AM=1 pionowo. W przykładzie AM=0, a I/D1--0=01, więc dane będą zapisywane do pamięci poziomo oraz adres będzie inkrementowany.

Ten tryb pracy ilustruje **rys. 6**. Dane do pamięci GRAM będą zapisywane od lewego dolnego rogu ekranu (poziomo) ku jego górze. Taki tryb pracy wyświetlacza jest związany z formą zapisu bitmapy w pliku. Bitmapa jest zapisywana od lewej dolnej części obrazu. Taki tryb pracy LCD ułatwia wyświetlenie bitmapy, bo wystarczy bezpośrednio wysyłać dane z pliku bitmapy prosto do wyświetlacza. W takim trybie wyświetlacz zapisuje dane do wyświetlenia tak, jak zostały one zapisane w pliku bitmapy.

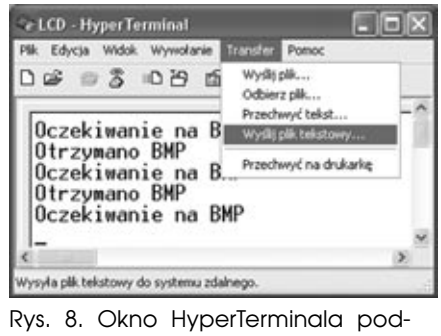
Bity LG2-0 rejestru R05h oraz CP15-0 rejestru R06h zostały wyzerowane, gdyż nie są wykorzystywane w przykładzie. Dotyczą one operacji logicznych i porównania na danych pamięci GRAM (na wyświetlanych pikselach).

Dalej, do rejestru kontrolującego cykl ramki o adresie R0Bh (tab.3) zapisywana zostaje wartość 0000h.

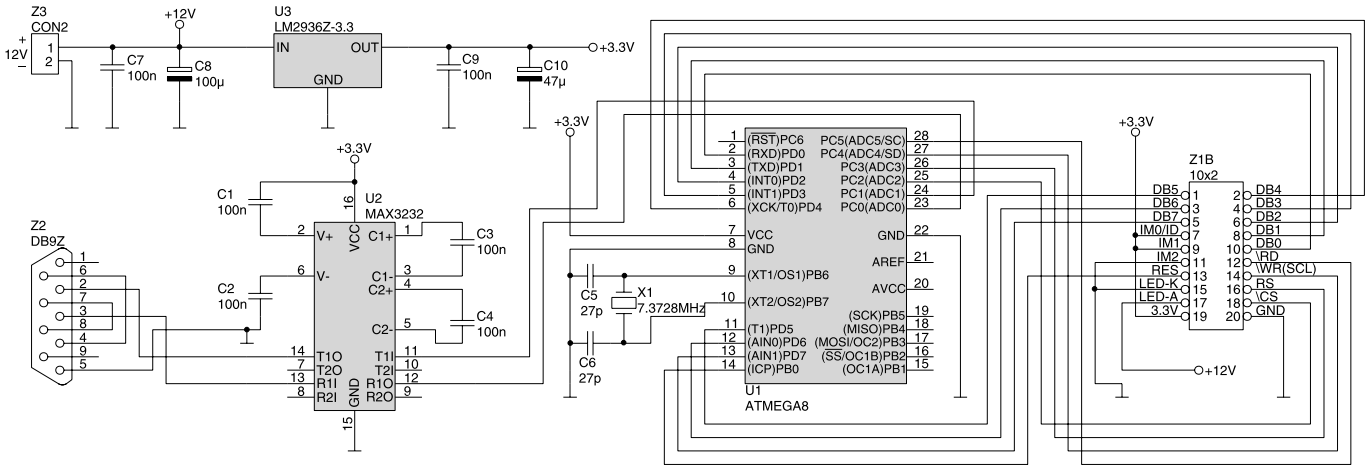
Bity RTN3-0 określają dodatkową liczbę cykli na jeden wiersz wyświetlacza, natomiast bity DIV1-0 określają stopień podziału częstotliwości oscylatora taktującej wewnętrzne bloki wyświetlacza. W przykładzie wszystkie bity zostały wyzerowane, więc nie ma dodatkowych cykli oraz podziału częstotliwości oscylatora.

Ekran wyświetlacza można podzielić na dwie części (dwa obrazy). W rejestrze R14h można zapisać współrzędne pierwszego ekranu, a w rejestrze R15h drugiego. Ponieważ ekran wyświetlacza został wykorzystany w całości do wyświetlenia bitmapy, więc wykorzystany został tylko rejestr R14h do określenia pozycji roboczej ekranu. Do tego rejestru została zapisana wartość 5300h. Bity SS17-10 tego rejestru określają początek pozycji ekranu 1, a SE17-10 koniec pozycji ekranu 1. Czyli będzie dostępny cały ekran wyświetlacza o rozdzielczości 96x64 pikseli.

Kolejne dane wysłane do rejestru R16h (tab.3) o wartości 5F00 i do rejestru R17h o wartości 3F00h, określają zakresy adresów pamięci GRAM odpowiadających wielkości ekranu wyświetlacza. Bity HSA7-0/HEA7-0 rejestru R16h określają poziomy początek i koniec okna (ekranu) dostępnego poprzez pamięć GRAM. Bity VSA7-0/VEA7-0 rejestru R17h określają pionowy początek i koniec okna (ekranu) dostępnego poprzez pamięć GRAM. Chodzi o to, że adres komórki pamięci będzie inkrementowany tylko



Rys. 8. Okno HyperTerminala podczas wymiany danych ze sterownikiem wyświetlacza



Rys. 9. Układ sterujący z interfejsem równoległym

z zakresów adresów zapisanych w rejestrach R16h i R17h. Wartość 5Fh oznacza linię o długości 96 pikseli, a wartość 3Fh 64 linie. Czyli adresowana będzie cała powierzchnia ekranu 96x64 pikseli.

Do kolejnego rejestru o adresie R20h zostają zapisane wartości 0000h. Jest to rejestr, który umożliwia maskowanie danych zapisywanych do GRAM. Ustawienie danego bitu maskuje ten bit. Ponieważ wszystkie bity zostały wyzerowane, więc jest brak maski przy zapisie danych do GRAM. Funkcję maskowania można na przykład wykorzystać do zmiany części koloru danego piksela. Następnie do rejestru kontrolnego wyświetlacza R07h (tab.3) zostaje zapisana wartość 0002h (00000000 00000010bin).

Bity LVE2-1 umożliwiają włączenie funkcji *Scroll* dla pierwszego ekranu jak i drugiego. W przykładzie bity te zostały wyzerowane, gdyż funkcje te nie są wykorzystywane.

Ustawienie bitu SPT powoduje wybranie drugiego ekranu, a wyzerowanie pierwszego. Ponieważ określony został tylko jeden ekran, bit ten został wyzerowany.

Kiedy bit B/W jest ustawiony, wtedy wszystkie piksele mogą zostać zapalone lub zgaszone zależnie od stanu bitu REV i niezależnie od zawartości pamięci GRAM. Przy ustawionym bicie REV wszystkie piksele będą zapalone, a przy wyzerowanym zgaszone. W przykładzie bity B/W i REV zostały wyzerowane.

Bity D1-0 umożliwiają włączenie wyświetlacza. Kiedy bit D1 jest ustawiony wyświetlacz jest włączony, a kiedy wyzerowany to wyłączony. Ustawienie bitu D0 włącza wyświetlenie danych zapisanych w pamięci GRAM na ekranie wyświetlacza.

Kolejna instrukcja wysłania do rejestru R07h tym razem wartości 0003h powoduje całkowite włączenie wyświetlacza. Rejestry R30h do R37h tworzą paletę odcieni kolorów. Dostępnych jest 16 5-bitowych palet. Dla każdej z palet można określić 24 poziomy odcieni. Można zostawić paletę domyślną, ale można także ją zmodyfikować, co zostało zrobione w przykładowym programie. Do palet zostały zapisane wartości od 1 do 24.

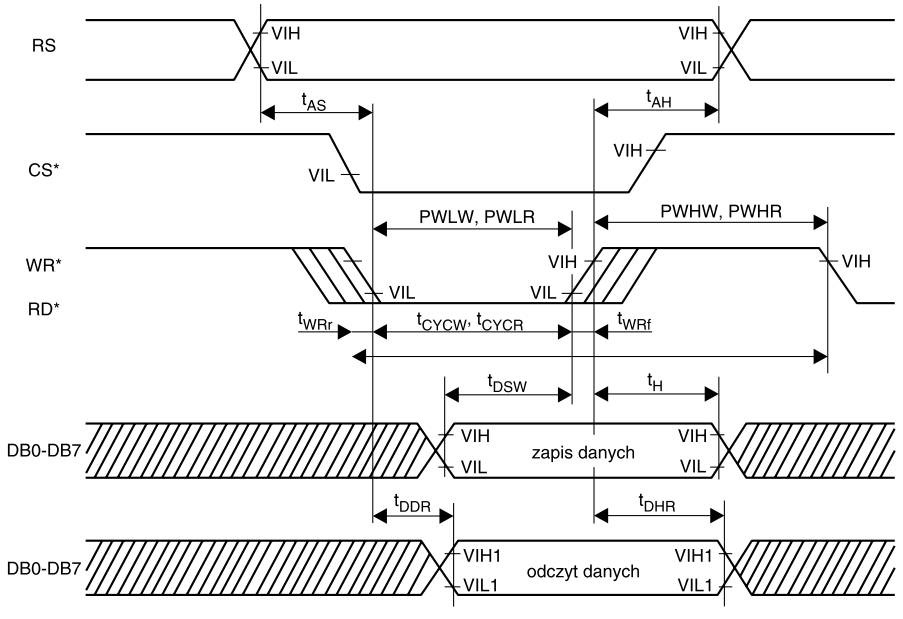
Przesłanie danych do wyświetlacza

Na tym etapie konfiguracja wyświetlacza się kończy i jest on gotowy do wyświetlenia danych. Program działa w nieskończonej pętli *Do-Loop*, w której pobiera dane z portu RS-232 i wysyła do wyświetlacza.

Na początku do komputera wysyłany jest komunikat oczekiwania na BMP (bitmapę). Plik bitmapy składa się z trzech części: nagłówka o długości 54 bajtów, który jest w przykładzie pomijany, opcjonalnej palety kolorów oraz danych obrazowych.

W przykładzie także zrezygnowano z palety kolorów umieszczonej w pliku bitmapy. Linie rysunku w bitmapie zapisywane są od dołu do góry, czyli tak jak będą zapisywane dane do wyświetlacza. Każdy punkt bitmapy zapisywany jest jako kolor RGB. W przypadku używanego wyświetlacza 16 bitowy kolor zapisywany jest jako R5G6B5, czyli kolor czerwony zapisywany jest na 5 bitach, zielony na 6, a niebieski na 5 bitach.

W pętli głównej programu do rejestru R21h wyświetlacza zostaje



Rys. 10. Przebiegi w interfejsie równoległym

List. 2. Przykład programu wyświetlającego wysłaną z komputera bitmapę

```
'Program przesyłający kolorową bitmapę 96x64 do
kolorowego wyświetlacza LCD
'Komunikacja rownolegla
'Marcin Wiązania
'marcin.wiazania@ep.com.pl

$regfile = „m8def.dat”
          'informuje kompilator o pliku

             'dyrektyw mikrokontrolera
$crystal = 7372800
          'informuje kompilator o

          'częstotliwości oscylatora

         'taktującego mikrokontroler
Config Portd = Output
' port D jako wyjsciowy
Config Portb = Output
'port B jako wyjsciowy
Config Portc = Output
'port C jako wyjsciowy

Open „COMC.1:38400,8,N,1” For Output As #1
'konfiguracja programowego interfejsu RS232 (linii
TXD), predkosc 38400
Open „COMC.0:38400,8,N,1” For Input As #2
'konfiguracja programowego interfejsu RS232 (linii
RXD), predkosc 38400

Declare Sub Zap_a_c(byval Adrh As Byte , Byval
Adrl As Byte , Byval Parh As Byte , Byval Parl
As Byte )       'procedura zapisu jednoczesnie
adresu i komendy
Declare Sub Zap_c(byval Comh As Byte , Byval Coml
As Byte)  'procedura zapisu parametrow
Declare Sub Zap_a(byval Adh As Byte , Byval Adl
As Byte)     'procedura zapisu adresu

Dim I As Byte
          'zmienna pomocnicza
Dim J As Byte
          'zmienna pomocnicza
Dim K As Word
          'zmienna licznikowa otrzymanych bajtow
z pliku BMP

Rd Alias Portc.5
          'alias do sygnalu odczytu Rd
Wr Alias Portc.4
          'alias do sygnalu zapisu Wr
Rs Alias Portc.3
            'alias do sygnalu rodzaju danych
(adres czy dane)
Cs Alias Portc.2
          'alias do sygnalu wyboru Cs
Rr Alias Portb.0
          'alias do sygnalu Rs (resetu)

Set Rd
          'ustawienie sygnalu Rd
Reset Rr
          'reset wyswietlacza
Set Wr
          'ustawienie sygnalu Wr
Set Rs
          'ustawienie sygnalu Rs
Set Cs
          'ustawienie sygnalu Cs
Waitms 20
          'czekaj 20 ms
Set Rr
          'koniec resetu

Call Zap_a_c(&H00 , &H00 , &H00 , &H01)
          'wlaczenie oscylatora wyswietlacza LCD
Waitms 20
          'czekaj 20 ms
Call Zap_a_c(&H00 , &H03 , &H12 , &H8C)
           'ustawienie parametrow zasilania
blokow wyswietlacza jak: BS2-0, BT1-0, DC2-0,
AP2-0, SLP
Call Zap_a_c(&H00 , &H0C , &H00 , &H00)
           'ustawienie parametrow zasilania
blokow wyswietlacza jak: VC2-0
Call Zap_a_c(&H00 , &H04 , &H1A , &HE0)
           'ustawienie parametrow kontrastu
wyswietlacza jak: VRCNT, VR4, VRON, CT6-1
Call Zap_a_c(&H00 , &H01 , &H02 , &H07)
          'ustawienie parametrow pracy drivera
sterujacego pixelami jak: CSFT, CMS, SGS, 4L,
NL4-0
Call Zap_a_c(&H00 , &H02 , &H00 , &H00)
          'ustawienie parametrow sygnalow steru-
jacych jak: RST, B/C, EOR, NW5-0
Call Zap_a_c(&H00 , &H05 , &H02 , &H10)
           'ustawienie parametrow odpowie-
dzialnych interpretowanie danych wejsciowych jak:
SPR1-0, HWM, I/D1-0, LG2-0
Call Zap_a_c(&H00 , &H06 , &H00 , &H00)
          'ustawienie regulowanego rezystora
komparatora jak: CP15-0
Call Zap_a_c(&H00 , &H0B , &H00 , &H00)
          'ustawienie cyklu ramki parametry:
DIV1-0, RTC3-0
Call Zap_a_c(&H00 , &H14 , &H53 , &H00)
          'ustalenie pozycji obrazu na ekranie
LCD parametry: SE17-10 i SS17-10
Call Zap_a_c(&H00 , &H16 , &H5F , &H00)
          'ustalenie poziomego adresu pamieci
RAM wyswietlacza
Call Zap_a_c(&H00 , &H17 , &H3F , &H00)
          'ustalenie pionowego aresu pamieci RAM
wyswietlacza
Call Zap_a_c(&H00 , &H20 , &H00 , &H00)
           'ustalenie maski zapisu danych do
pamieci RAM (brak zdefiniowanej maski)
Call Zap_a_c(&H00 , &H07 , &H00 , &H02)
          'ustalenie parametrow kontrolnych
wyswietlacza jak: VLE1-0, SPT, B/W i REV
Call Zap_a_c(&H00 , &H07 , &H00 , &H03)
          'ustalenie parametrow kontrolnych
wyswietlacza jak: D1-0 (właczenie LCD)
Call Zap_a_c(&H00 , &H30 , 3 , 1)
          'konfiguracja patety odcieni
Call Zap_a_c(&H00 , &H31 , 7 , 5)
          'dalsza konfiguracja patety odcieni
Call Zap_a_c(&H00 , &H32 , 9 , 8)
          'dalsza konfiguracja patety odcieni
Call Zap_a_c(&H00 , &H33 , 12 , 11)
          'dalsza konfiguracja patety odcieni
Call Zap_a_c(&H00 , &H34 , 14 , 13)
          'dalsza konfiguracja patety odcieni
Call Zap_a_c(&H00 , &H35 , 16 , 15)
          'dalsza konfiguracja patety odcieni
Call Zap_a_c(&H00 , &H36 , 19 , 17)
          'dalsza konfiguracja patety odcieni
Call Zap_a_c(&H00 , &H37 , 24 , 22)
          'dalsza konfiguracja patety odcieni


Do
          'petla glowna programu
  Print #1 , „Oczekiwanie na BMP”
          'komunikat wyslany do terminala
  K = 0
          'wyzerowanie zmiennej K
  Call Zap_a_c(&H00 , &H21 , &H3F , &H00)
'ustawienie adresu poczatkowego pamieci GRAM,
ktory bedzie automatycznie inkrementowany
  I = 0
          'wyzerowanie zmiennej I
  Call Zap_a(&H00 , &H22)
              'adres komendy zapisu danych do
pamieci GRAM
  Do
          'petla wykonywana az I = 70
    Incr I
          'zwieszenie o 1 wartości I
    J = Waitkey(#2)
'oczekiwanie na odebranie bajta danych z portu
RS232 do zmiennej J
  Loop Until I = 70
          'jesli I = 70 to koniec petli
  Do
          'petla wykonywana az K = 6145
    J = Waitkey(#2)
'oczekiwanie na odebranie bajta danych z portu
RS232 do zmiennej J
    I = Waitkey(#2)
'oczekiwanie na odebranie bajta danych z portu
RS232 do zmiennej I
    Call Zap_c(i , J)
              'wywolanie procedury zapisu dwoch
kolejnych pixeli
    Incr K
          'zwiekszenie o 1 zmiennej K
  Loop Until K = 6145
          'jesli K=6145 to koniec petli
  Print #1 , „Otrzymano BMP”
           'wyslanie to terminala komunikatu o
otrzymaniu calej bitmapy
Loop
          'koniec petli glownej programu
End
          'koniec programu

Sub Zap_a_c(byval Adrh As Byte , Byval Adrl
As Byte , Byval Parh As Byte , Byval Parl As
Byte)       'procedura zapisu jednoczesnie adresu
oraz komendy
   Call Zap_a(adrh , Adrl)
          'wywolanie procedury zapisu adresu
   Call Zap_c(parh , Parl)
          'wywolanie procedury zapisu parametrow
(danych)
End Sub

Sub Zap_a(byval Adh As Byte , Byval Adl As Byte)
'procedura zapisu adresu komendy
  Reset Rs
          'linia Rs wyzerowana - zapis adresu
  nop
          'opoznienie o jeden cykl
  Reset Cs
           'wybor komunikacji z LCD - linia CS
wyzerowana
  nop
          'opoznienie o jeden cykl
  Reset Wr
          'zerowanie linii zapisu Wr
  Portd = Adh
           'wystawienie na port bardziej znacza-
cego slowa adresu
  nop
          'opoznienie o jeden cykl
  Set Wr
           'zapis do LCD bajtu wystawionego na
liniach portu D
  nop
          'opoznienie o jeden cykl
  Reset Wr
          'zerowanie linii zapisu Wr
  Portd = Adl
           'wystawienie na port mniej znaczacego
slowa adresu
  nop
          'opoznienie o jeden cykl
  Set Wr
           'zapis do LCD bajtu wystawionego na
liniach portu D
  nop
          'opoznienie o jeden cykl
  Reset Cs
           'wybor komunikacji z LCD - linia CS
wyzerowana
  nop
          'opoznienie o jeden cykl
  Reset Wr
          'zerowanie linii zapisu Wr
  Portd = Comh
           'wystawienie na port bardziej znacza-
cego bajta danej
  nop
          'opoznienie o jeden cykl
  Set Wr
           'zapis do LCD bajtu wystawionego na
liniach portu D
  nop
          'opoznienie o jeden cykl
  Reset Cs
           'wybor komunikacji z LCD - linia CS
wyzerowana
  nop
          'opoznienie o jeden cykl
  Reset Wr
          'zerowanie linii zapisu Wr
  Portd = Coml
           'wystawienie na port mniej znaczacego
bajta danej
  nop
          'opoznienie o jeden cykl
  Set Wr
           'zapis do LCD bajtu wystawionego na
liniach portu D
  nop
          'opoznienie o jeden cykl
  Set Cs
           'ustawienie sygnalu Cs - koniec
komunikacji z LCD
End Sub
```

zapisana wartość 3F00h, która jest początkiem adresu pamięci GRAM automatycznie inkrementowanego. Wartość adresu 3F00h wskazuje na początek lewego dolnego rogu ekranu.

Następnie w programie poprzez wykonanie procedury *Zap_a* zostaje wybrany tylko rejestr R22h, który jest rejestrem zapisu danych do pamięci GRAM pod określony wcześniej adres. Zapis określonej danej do pamięci GRAM będzie powodować zapalenie odpowiadającego jej piksela w wybranym kolorze.

Kolor piksela jest zapisywany w formie R5G6B5 dla 16-bitowej głębi kolorów co ilustruje **rys. 7**.

W pierwszej kolejności w programie jest pomijany nagłówek bitmapy, po którym następują dane obrazu, które należy przesłać do wyświetlacza. Program po odebraniu dwóch bajtów określających kolor (jedno słowo) wysyła je od razu do wyświetlacza za pośrednictwem procedury *Zap_c*. Po otrzymaniu wszystkich danych obrazowych wysyłany jest komunikat o otrzymaniu obrazu i następuje oczekiwanie programu na przesłanie kolejnej bitmapy, która zostanie wyświetlona. Adres wskazujący na komórki w pamięci GRAM jest automatycznie inkrementowany.

Przykładowe bitmapy przystosowane do wyświetlania na wyświetlaczu opisanym w artykule

Jak widać wyświetlenie danych na takim wyświetlaczu jest bardzo proste, gdyż wystarczy tylko do pamięci GRAM zapisać kolor piksela.

W ramach przykładu przygotowane zostały bitmapy bez kompresji o rozdzielczości 96x64 i zapisie kolorów w formie R5G6B5. Tak przygotowane bitmapy można było bezpośrednio wysyłać do wyświetlacza bez żadnej konwersji. Do wysłania bitmapy posłużył *Hyper Terminal* dostępny w Windows (**rys. 8**). Plik bitmapy został wysłany przez wykonanie polecenia *Wyślij plik tekstowy* z menu *Transfer*.

Połączenie z wyświetlaczem przez interfejs równoległy

Jeśli będzie potrzebne szybkie wysyłanie danych do wyświetlacza, na przykład przy realizacji animacji, to należy wykorzystać do przesyłania danych równoległy interfejs wyświetlacza.

Na **rys. 9** przedstawiono przykład dołączenia wyświetlacza do mikrokontrolera z wykorzystaniem interfejsu równoległego. Linie IM0, IM1 i IM2 powinny zostać skonfigurowane zgodnie z tab. 2. Komunikacja z wyświetlaczem w sposób równoległy odbywa się jak to przedstawiono na **rys. 10.**

Sygnał RS określa czy wysyłany jest adres rejestru wyświetlacza, czy dana zapisywana w rejestrze. Przy komunikacji równoległej z wyświetlaczem posłużyłem się tym samym przykładem co przy komunikacji szeregowej. Na **list. 2** przedstawiony został przykład programu wyświetlającego wysłaną z komputera bitmapę na wyświetlaczu z komunikacją równoległą. Ponieważ linie sprzętowego interfejsu RS232 zostały wykorzystane w innych celach, interfejs RS232 został zrealizowany programowo. Komunikacja z komputerem odbywa się z prędkością 38400 bodów. Analizę równoległej komunikacji z wyświetlaczem w tym programie pozostawiam czytelnikowi.

Podsumowanie

W zaprezentowanym wyświetlaczu nie ma generatora znaków. Jeśli będą wyświetlane znaki alfanumeryczne, to należy je wcześniej przygotować i umieścić w jakiejś tablicy. We własnych aplikacjach można także wykorzystać szeregowy synchroniczny interfejs 4 przewodowy, w którym dodatkową linią jest linia RS, od której zależy czy wysyłany będzie adres, czy wartość rejestru. Zaprezentowany wyświetlacz posiada wiele dodatkowych funkcji, jak funkcję maskowania, porównania, Scroll itp. Po dodatkowe informacji odsyłam do dokumentacji układu HD66768. Jak zostało pokazane obsługa kolorowego graficznego wyświetlacza nie musi uchodzić za bardzo trudną i niemożliwą do wykonania we własnych opracowaniach. Obsługa niczym nie różni się od obsługi typowego graficznego wyświetlacza monochromatycznego. W przypadku wyświetlaczy kolorowych zyskuje się wiele, bo występują nie dwa kolory, ale nawet 65000 kolorów. Stosowanie takich dosyć tanich kolorowych wyświetlaczy w znaczący sposób podniesie walory użytkowe konstrukcji.

Marcin Wiązania, EP
marcin.wiazania@ep.com.pl