

Sterowanie wskaźnikami ciekłokrystalicznymi

Sterownik wyświetlacza graficznego, część 2

W drugiej części artykułu poświęconego sterowaniu graficznego wyświetlacza LCD prezentujemy listę rozkazów kontrolera HD61830B oraz algorytm rysowania figur geometrycznych.

Programowanie układu HD61830B

Układ HD61830B programowany jest za pomocą rejestru instrukcji (IR) oraz ogólnego rejestru danych wejścia/wyjścia (DIOR). Dostępnych jest 13 dwubajtowych rozkazów. Rozkazy do kontrolera przesyłane są w dwu kolejnych cyklach. Pierwszy zapis do rejestru IR specyfikuje kod rozkazu oraz numer jednego z 13 wewnętrznych rejestrów danych. W drugim cyklu zapisu do (lub odczytu z) rejestru DIOR informacja zostaje przesłana do (odczytana z) wybranego rejestru wewnętrznego. Pełna lista rozkazów kontrolera HD61830B zestawiona jest w tab. 1. W dalszej części przedstawiony jest szczegółowy opis poszczególnych rozkazów.

Mode control. Kod 0x00

Rozkaz „Mode control“ ustala sposób pracy wskaźnika (D1=0: tryb tekstowy, D1=1: tryb graficzny). W trybie tekstowym ustawia się atrybuty kursora (D2=0: kursor wyłączony, D2=1: kursor włączony), mruganie znaku pod kursorem (D3=0: znak nie mrugający, D3=1: znak mrugający), wewnętrzny (D0=0) lub zewnętrzny (D0=1) generator znaków. Zarówno w trybie tekstowym jak i graficznym wskaźnik można zgasić (D5=0: wskaźnik włączony, D5=1: wskaźnik wyłączony). Bit D4 umożliwia konfigurację sterownika wskaźnika w której wykorzystuje się dwa równoległe pracujące kontrolery HD61830. W przypadku stosowania pojedynczego kontrolera D4=1.

Set character pitch. Kod 0x01

Rozkaz „Set character pitch“ dla trybu tekstowego ustala rozmiary, wraz z marginesami, drukowanego znaku. Wartość Vp, istotna jedynie w trybie tekstowym, oznacza wysokość znaku a Hp jego szerokość. Możliwe są jedynie trzy wartości Hp:

Hp=6, dla szerokości 6 punktów,
Hp=7, dla szerokości 7 punktów,
Hp=8, dla szerokości 8 punktów.

W trybie graficznym Hp określa ile bitów z bajtu ma być wyświetlanych (Hp=8: oznacza wszystkie bity).

Set number of characters.

Kod 0x02

Rozkaz „Set number of characters“ ustala ilość znaków wyświetlanych w linii w trybie tekstowym lub ilość bajtów (1 bit odpowiada 1 punktowi) przypadających na jedną linię w trybie graficznym.

Set number of time division.

Kod 0x03

Parametr „Time division“ jest odwrotnością wielkości duty ratio (cykl wypełnienia) charakteryzującą konkretny wyświetlacz. Dla wskaźnika LMG640X duty ratio wynosi 1/128, stąd Nx=128.

Set cursor position. Kod 0x04

Parametr „Cursor position“ ważny jedynie w trybie tekstowym ustala numer wiersza w matrycy znaku gdzie wyświetlany będzie kursor. Dla przykładu w matrycy 5x7 kursor wyświetlany bezpośrednio pod znakiem ustala się przez Cp=8.

Set display start low address.

Kod 0x08

Set display start high address.

Kod 0x09

Rozkazy „Set display address“ ustalają odwzorowanie lewego górnego rogu wskaźnika (pozycja Home) na pamięć RAM ekranu. W trybie graficznym adres ten składa się 16 bitów. W trybie tekstowym ważne są jedynie bity D3..D0 starszego bajtu i wszystkie bity młodszego.

Set cursor low address.

Kod 0x0a

Set cursor high address.

Kod 0x0b

Rozkazy „Set cursor address“ ustalają adres kursora w pamięci RAM. Kolejne zapisy (lub odczyty z) pamięci dokonywane będą do (lub z) lokacji ustalonej przez położenie kursora. Producent zaleca aby adres kursora był ustawiany w kolejności: najpierw młodszy a po tym starszy bajt.

Write display data. Kod 0x0c

Read display data. Kod 0x0d

Rozkazy „Write/Read display data“ służą do zapisu/odczytu kodu znaku (w trybie tekstowym) lub bajtu (w trybie graficznym). Po wykonaniu rozkazu adres kursora jest automatycznie zwiększany o 1.

Tab. 1. Lista rozkazów kontrolera HD61830B.

rozkaz	IR	DIOR	D7	D6	D5	D4	D3	D2	D1	D0
Mode control	0x00	0	0	dis	ms	blk	cur	gr	cg	
Set character pitch	0x01	(Vp-1)	0	(Hp-1)						
Set number of chars	0x02	0	(Hn-1)							
Set number of divs	0x03	0	(Nx-1)							
Set cursor pos	0x04	0	0	0	0	(Cp-1)				
Set disp addr low	0x08	(address)								
Set disp addr high	0x09	(address)								
Set cursor addr low	0x0a	(address)								
Set cursor addr high	0x0b	(address)								
Write display data	0x0c	(char/data)								
Read display data	0x0d	(char/data)								
Clear bit	0x0e	0	0	0	0	0	(Nb-1)			
Set bit	0x0f	0	0	0	0	0	(Nb-1)			
Read busy flag										

Clear bit. Kod 0x0e**Set bit. Kod 0x0f**

Rozkazy „Set/Clear bit” umożliwiają manipulację pojedynczymi punktami na ekranie. Wartość Nb ustala pozycję bitu w bajcie ($Nb=1$ odpowiada D0, $Nb=8$ odpowiada D7). Po wykonaniu rozkazu adres kursora jest automatycznie zwiększany o 1.

Read busy flag

Rozkaz „Read busy flag” umożliwia sprawdzenie stanu zajętości kontrolera. Rozkaz wykonywany jest przez bezpośrednio odczytanie bajtu z rejestru rozkazów IR. Bit D7=1 oznacza, że kontroler jest w trakcie wykonywania poprzedniego rozkazu i nie może otrzymać ani następnego rozkazu ani danej będącej częścią rozkazu.

Konfiguracja wskaźnika LMG640X

Jak widać z powyższego opisu konfiguracja wskaźnika sprowadza się do obliczenia kilku parametrów a następnie wpisania ich do rejestrów wewnętrznych kontrolera. Dokonajmy wszystkich obliczeń zarówno dla trybu tekstowego jak i graficznego. Aby ustawić włączony wskaźnik w trybie tekstowym, z wewnętrznym generatorem znaków, z mrugającym i włączonym kursorem należy ustawić słowo kontrolne na 0x3C. Ustawienie trybu graficznego z wyłączonym kursorem uzyskamy programując rejestr *mode* na 0x32.

Załóżmy, że w trybie tekstowym stosować będziemy matrycę 6x8 (znak 5x7 oraz jednopunktowy margines), wtedy $Vp=8$, $Hp=6$. Jeśli przyjmijemy dodatkowo, że kursor umieścimy w ostatniej linii matrycy znaków to $Cp=8$. Jeśli w trybie graficznym mierzymy odwzorowywać każdy bit bajtu z pamięci RAM musimy przyjąć $Hp=8$.

Przyjmując szerokość znaku jako $Vp=6$ to w linii możemy wyświetlić $Hn=240/Hp=40$ znaków. W trybie graficznym na jedną linię będzie przypadać $Hn=240/Hp=30$ bajtów. Wartość Nx dla wskaźnika LMG640X z danych katalogowych producenta przyjmujemy jako $Nx=128$. Jako lokację *Home* najwygodniej będzie przyjąć początek pamięci RAM to jest adres 0. W tab. 2 zebrane są przykładowe parametry konfiguracji zarówno dla trybu tekstowego jak i graficznego.

Trochę rachunku różniczkowego

Nawet najwspanialszy wskaźnik graficzny będzie mało użyteczny jeśli nie będziemy w stanie generować na nim obrazów graficznych. Aby móc praktycznie korzystać z możliwości graficznych potrzebne są przynajmniej funkcje rysujące pojedynczy punkt, linię, prostokąt i okrąg lub elipsę.

Podstawową funkcją graficzną biblioteki obsługi wskaźnika LMG640X jest funkcja *lcgputpixel()* umożliwiająca włączenie lub zgaszenie dowolnego punktu na wskaźniku.

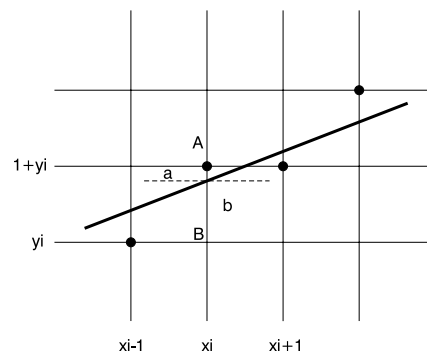
Punktem wyjścia umożliwiającym analizę działania wszystkich funkcji graficznych jest sposób w jaki kontroler HD61830B odwzorowuje wewnętrzny RAM na ekran. Lewy górny róg ekranu odpowiada najstarszemu bitowi pierwszej komórki pamięci RAM. Kolejne bity odwzorowują kolejne punkty na ekranie. Na każdy wiersz wskaźnika w trybie graficznym przypada 30 kolejnych bajtów. Logiczną organizację ekranu przyjmujemy podobnie jak na ekranie komputera PC jako czwartą ćwiartkę układu współrzędnych. Oś X skierowana będzie w prawo zaś oś Y od punktu HOME w dół. Przy tak przyjętym układzie odniesienia punkt o współrzędnej (x, y) odpowiadać będzie bajtowi o adresie: $loc=(240*y+x)/8$ i bitowi w bajcie *loc* o numerze $pixel=(240*y+x)\%8$. Mając wartości *loc* i *pixel*, rozkazami „Set cursor address” ustalamy adres bajtu a rozkazami „Set/Clear bit” włączamy lub gasimy żądany punkt na ekranie.

Mając możliwość kontroli dowolnego punktu na ekranie możemy już kreślić linie i okręgi. Rysowanie linii pionowych i poziomych jest trywialne. Wystarczy w pętli zmieniać jedną z współrzędnych i rysować kolejne punkty. Problem pojawia się w przypadku gdy zamierzamy narysować linię pod dowolnym kątem. Oczywiście można tę linię rysować wprost korzystając z ogólnego wzoru funkcji liniowej: $y=a*x+b$, gdzie a i b są stałymi a x zmienną i y oczekiwaną wartością funkcji. Niestety powyższy sposób jest mało efektywny i w rezultacie, ze względu na zaokrąglenia, da nam poszarpaną, a co najważniejsze nieciągłą linię. Znacznie efektywniejszym i elegantszym rozwiązaniem jest użycie algorytmu numerycznego.

Podstawą wszelkich metod numerycznych jest zasada obliczania kolejnej wartości funkcji w oparciu o wartość poprzednią i ustalony mały krok:

1. ustalamy wartości początkowe (krok, zmienna, wartość),
2. obliczamy kolejną wartość,
3. zwiększamy zmienną o wartość kroku,
4. przechodzimy do punktu 2.

W przypadku kreślenia figur na ekranie krok będzie miał wartość 1 (1 punkt) a współrzędne będą się zmieniały w zakresie $<0, x, 240$ i $<0, y, 128$). Zauważmy, że ekran możemy potraktować jako prostokątną siatkę z punktami na ekranie odpowiadającymi przecięciom linii siatki. Na tak rozumiany ekran możemy nałożyć hipotetyczną linię ($y=a*x+b$) do wykreślenia. Oczywiście jest, że na ekranie mogą pojawić się jedynie punkty wynikające z przecięcia się linii siatki a nie wynikające z równania linii. Prezentowany algorytm Bresenham’a działa w ten sposób, że obliczając wartość funkcji (y) dla kolejnego punktu (x)



Rys. 4.

dokonywane jest porównanie z najbliższymi punktami siatki. W zależności od odległości od siatki punkt rysowany jest bliżej górnego (A) lub dolnego (B) wiersza tak jak pokazane jest to na rys. 3. Równanie linii $y=a*x+b$ można przekształcić do postaci numerycznej: $y=(dy/dx)*x+b$, gdzie dy/dx oznacza tangens nachylenia linii ($dy=y_2-y_1$ i $dx=x_2-x_1$). Różnicę odległości pomiędzy prawdziwą wartością funkcji w punkcie x ($(dy/dx)*x+b$) a punktami A ($y+1$) i B (y) można wyrazić jako: $d(x)=a-b=2*(dy/dx)*x - 2*y - 1$.

Istotą algorytmu Bresenham’a nie jest obliczanie bezwzględnej wartości odległości ($d(x)$) od siatki w punkcie x ale różnicy w stosunku do odległości obliczonej w poprzednim punkcie ($d(i)-d(i-1)$). Jeśli względna różnica jest większa od 0 punkt umieszczamy w A w przeciwnym razie w B. W podobny sposób działa algorytm rysowania elipsy (lub okręgu jeśli przyjmijemy równe półosie). Co więcej ze względu na symetrię elipsy wystarczy obliczyć jedynie współrzędne dla pierwszej ćwiartki, a następnie symetrycznie w pozostałych ćwiartkach umieścić punkty na ekranie. Mimo iż algorytmy rysowania linii i elipsy wyglądają na skomplikowane ich praktyczna realizacja w funkcjach *lcline()* i *lcelipse()* jest stosunkowo prosta.

Przykład programu obsługi wskaźnika

Obsługa wskaźnika LMG640X sprowadza się do jego inicjacji do trybu tekstowego lub graficznego. Po inicjacji wskaźnik gotowy jest do pracy. Przykład programu przedstawiony w list. 2 (EP10/99) rysuje na ekranie dwa prostokąty, dwie linie ukośne po czym rysuje wskaźnik elipsę i okrąg.

Janusz J. Młodzianowski

Karta katalogowa (PDF) sterownika HD61830 jest dostępna pod adresem <http://www.ep.com.pl/ftp/lcdgraf.pdf>.

Tabela 2. Przykładowe parametry konfiguracyjne wskaźnika LMG640X.		
rozkaz	tryb tekstowy	tryb graficzny
Mode control	0x3C	0x32
Set character pitch	0x75	0x77
Set number of chars	0x27	0x1D
Set number of divs	0x7F	0x7F
Set cursor pos	0x07	-
Set disp addr low	0x00	0x00
Set disp addr high	0x00	0x00