

Sterowanie graficznych wyświetlaczy z telefonów komórkowych firmy Nokia, część 2

W drugiej części artykułu przedstawiamy protokół komunikacyjny sterownika PCD8544, polecenia nim sterujące oraz napisaną w assemblerze procedurę inicjującą sterownik po włączeniu zasilania.

Protokół transmisji

Kontroler może interpretować wysyłane do niego informacje na dwa sposoby. Jeśli linia D/C jest w stanie wysokim, to przyjmowane dane są wpisywane do pamięci RAM kontrolera. Jeśli linia D/C jest w stanie niskim, to wysłany bajt interpretowany jest jako komenda sterująca. Kolejne bajty danych wysyłane są do wyświetlacza poprzez linię SDIN w takt sygnału zegarowego podanego na linię SCLK (rys. 8). Poszczególne bity wysyłane są w kolejności od najstarszego do najmłodszego. Stan linii SDIN jest sprawdzany podczas narastającego zbocza sygnału SCLK, a stan linii D/C jest sprawdzany podczas odbioru ostatniego bitu należącego do jednego bajtu danych. Jeśli linia SCE jest w stanie wysokim, to wyświetlacz ignoruje wszelkie dane pojawiające się na wejściu SDIN, a interfejs szeregowy kontrolera pozostaje w trybie obniżonego poboru mocy. Opadające zbocze na linii SCE inicjuje pracę interfejsu wejściowego i jest zarazem wyznacznikiem początku transmisji bloku danych. W jednym bloku mogą być transmitowane zarówno komendy, jak i dane, w dowolnej kolejności (rys. 9).

Komendy sterujące

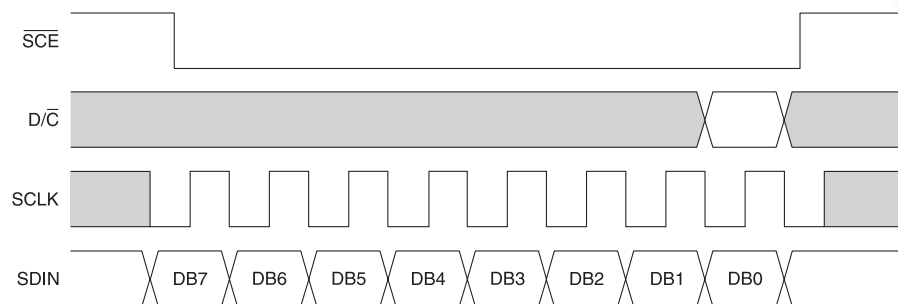
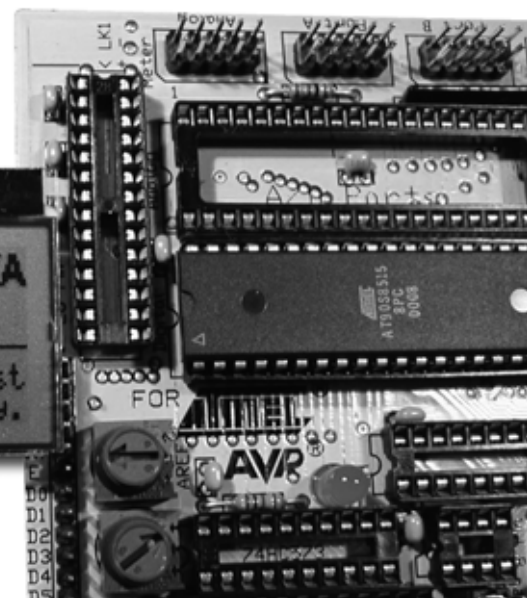
W tab. 4 pokazano pełny zestaw komend sterujących pracą wyświetlacza opartego na kontrolerze PCD8544. Istnieją dwa zestawy komend: standardowy i rozszerzony. Aby skorzystać z rozszerzonego zestawu komend, należy najpierw wysłać do wyświetlacza komendę *Function set* z ustawionym bitem H - czyli DB0, a następnie wysłać dowolną ilość komend trybu rozszerzonego. Żeby powrócić do standardowego zestawu, należy wysłać komendę *Function set*, lecz z wyzerowanym bitem H. Zapis danych do pamięci RAM wyświetlacza (D/C = 1) jest możliwy zawsze, niezależnie od wybranego zestawu komend.

Niektóre wersje wyświetlaczy (np. LPH-7779 ze stykami w postaci złocistych padów, a nie sprężystych blaszek) wykorzystują jako kontroler układy firmy Seiko-Epson, które są bardzo zbliżone do kontrolerów

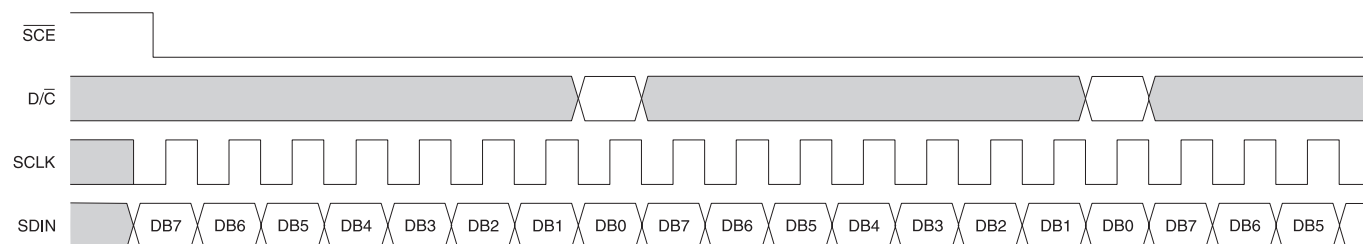
PCD8544. Większość komend sterujących jest identyczna, lecz niektóre komendy nie działają na kontrolerach Epsona. Przykładem jest tryb negatywny lub brak możliwości regulacji kontrastu - wyświetlacze z kontrolerami Epsona mają optymalnie ustawiony kontrast w fazie ich produkcji.

Ustawienie kontrastu i kompensacji temperaturowej

Aby treść wyświetlana na matrycy LCD była widoczna, należy ustawić odpowiednią wartość napięcia zasilającego matrycę, co jest równoznaczne z ustawieniem optymalnego kontrastu wyświetlacza. Prze-



Rys. 8. Sposób transmisji do wyświetlacza jednego bajtu danych



Rys. 9. W ten sposób można wysłać do kontrolera wyświetlacza kilka bajtów danych

widziano do tego komendę *Set Vop* dostępną w rozszerzonym zestawie komend. Teoretycznie można wybrać jedną ze 128 możliwych wartości, lecz praktycznie użyteczny zakres regulacji kontrastu wynosi od 32 do 88, czyli heksadecymalnie od 0x20 do 0x58. Należy pamiętać, że komenda *Set Vop* ma ustawiony bit DB7, więc do podanych wartości należy przed wysłaniem do wyświetlacza dodać 128 oraz wcześniej przełączyć się na rozszerzony zestaw komend. Nie należy również przekraczać podanego górnego zakre-

Tab. 4. Wykaz poleceń sterujących pracą kontrolera PCD8544

Instrukcja	DC	Bajtkomendy								Opis	
		DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
H=0 lub 1 - oba zestawy komend											
NOP	0	0	0	0	0	0	0	0	0	0	Brak akcji
Function set	0	0	0	1	0	0	PD	V	H		Kontrola trybu <i>Power Down</i> Tryb adresowania (V) Rozszerzone komendy (H)
Write data	1	D7	D6	D5	D4	D3	D2	D1	D0		Zapis danych do pamięci RAM
H=0 - podstawowy zestaw komend											
Zarezerwowana	0	0	0	0	0	0	1	X	X		Nie używać
Display Control	0	0	0	0	0	1	D	0	E		Konfiguracja wyświetlacza
Zarezerwowana	0	0	0	0	1	X	X	X	X		Nie używać
Set Y□address	0	0	1	0	0	0	Y2	Y1	Y0		Ustawienie licznika wierszy (Y = 0...5)
Set X□address	0	1	X6	X5	X4	X3	X2	X1	X0		Ustawienie licznika kolumn (X = 0...83)
H=1 - rozszerzony zestaw komend											
Zarezerwowana	0	0	0	0	0	0	0	0	1		Nie używać
Zarezerwowana	0	0	0	0	0	0	0	1	X		Nie używać
Temperature control	0	0	0	0	0	0	1	TC1	TC0		Ustawienie współczynnika korekcji temperaturowej (Tc = 0...3)
Zarezerwowana	0	0	0	0	0	1	X	X	X		Nie używać
Bias system	0	0	0	0	1	0	BS2	BS1	BS0		Ustawienie trybu multipleksowania
Zarezerwowana	0	0	1	X	X	X	X	X	X		Nie używać
Set Vop	0	1	Vop6	Vop5	Vop4	Vop3	Vop2	Vop1	Vop0		Ustawienie napięcia zasilania matrycy, czyli kontrastu

Legenda:

X□ - bit nieistotny

PD - 1□ = tryb *Power Down*, 0□ = kontroler aktywny

V□ - 1□ = adresowanie w□ trybie pionowym, 0□ = adresowanie w□ trybie poziomym

H□ - 1□ = wybór rozszerzonego zestawu komend, 0□ = wybór podstawowego zestawu komend

D, E□ - 00 = wyświetlacz wygaszony, 10 = tryb standardowy, 01 = test załączający wszystkie piksele, 11 = tryb negatywny

TC1, TC0 - Wartość korekcji temperaturowej - standardowa wartość 2□ (cała komenda to 0x06)

BS2...BS0 - Dla opisywanych wyświetlaczy powinno być 3□ (cała komenda to 0x13)

Vop6...Vop0 - Standardowa wartość to 0x48 (cała komenda to 0xC8)

su wartości, ponieważ przy niskich temperaturach otoczenia możemy przekroczyć maksymalne napięcie zasilania matrycy i uszkodzić matrycę lub kontroler - czyli cały wyświetlacz.

Konieczność wprowadzenia dodatkowej automatycznej korekcji napięcia V_{LCD} w funkcji temperatury podyktowana jest zmienną czułością ciekłego kryształu w zależności od temperatury oraz niezerowym współczynnikiem temperaturowym samego układu kontrolera. Układ PCD8544 posiada możliwość wyboru jednej z 4 predefiniowanych charakterystyk kompensacji o nachyleniu charakterystyki od 1 mV/K do 24 mV/K. W większości zastosowań wystarczy ustawienie trzeciej charakterystyki (17 mV/K) poprzez ustawienie bitów TC1 i TC0 na 1,0, co wymaga wysłania w trybie komend rozszerzonych bajtu o wartości 0x06. Oczywiście jeśli się zdarzy, że nasz wyświetlacz będzie wykazywał tendencje do zmiany kontrastu spowodowanej zmianami temperatury otoczenia, możemy spróbować zminimalizować ten efekt poprzez ustawienie jednej z 3 pozostałych charakterystyk kompensacji.

Procedura inicjalizacji wyświetlacza

Bezpośrednio po załączeniu zasilania wyświetlacza zawartość rejestrów wewnętrznych kontrolera oraz jego pamięci graficznej RAM jest nieokreślona. Dlatego też bardzo ważną czynnością jest doprowadzenie do wyświetlacza prawidłowego sygnału RESET w czasie maksimum 30 ms po załączeniu zasilania, o poziomie niskim i czasie jego trwania minimum 100 ns. Według materiałów firmy Philips brak poprawnego zerowania może nawet doprowadzić do uszkodzenia wyświetlacza. Po wyzerowaniu kontrolera znajduje się on w trybie *Power Down*, ma ustawione adresowanie poziome, wyświetlacz jest wygaszony (D=E = 0), a przetwornica generująca V_{LCD} jest wyłączona. Zawartość pamięci RAM nie jest zerowana.

Żeby poprawnie zainicjować wyświetlacz, należy wykonać następującą sekwencję:

- ustawić linie SCE oraz D/C w stan niski,
- wysłać bajt 0x21 - przełączenie na komendy rozszerzone,
- wysłać bajt 0xC8 - ustawienie Vop, czyli kontrastu (może wymagać zmiany wartości kontrastu),
- wysłać bajt 0x06 - ustawienie współczynnika korekcji temperaturowej,
- wysłać bajt 0x13 - ustawienie współczynnika multipleksowania 1:48,
- wysłać bajt 0x20 - przełączenie na komendy standardowe, adresowanie poziome (0x22 dla adresowania pionowego),
- wysłać bajt 0x0C - Display Control - tryb standardowy,
- wysłać bajt 0x40 - zerowanie licznika wierszy,
- wysłać bajt 0x80 - zerowanie licznika kolumn,
- ustawić linie D/C w stan wysoki,

List. 1. Procedura inicjująca interfejs SPI mikrokontrolera AVR oraz podłączony do niego wyświetlacz

```

lcd_init:    sbi        CS_PORT,CS_PIN        ;CS = H
            sbi        CS_PORT-1,CS_PIN    ;CS port jako wyjście
            sbi        DC_PORT-1,DC_PIN    ;C/D jako wyjście
            call       SPI_init            ;inicjalizacja interfejsu SPI

            cbi        DC_PORT,DC_PIN      ;CD = L
            ldi        r24,0x21            ;przełącz na rozszerzone komendy
            rcall     lcd_wr
            ldi        r24,0xC8            ;ustaw Vop (kontrast)
            rcall     lcd_wr
            ldi        r24,0x06            ;ustaw współczynnik temperaturowy
            rcall     lcd_wr
            ldi        r24,0x13            ;ustaw multipleks na 1:48
            rcall     lcd_wr
            ldi        r24,0x20            ;przełącz na standardowe komendy,
            rcall     lcd_wr                ;oraz włącz adresowanie poziome
            ldi        r24,0x0C            ;włącz tryb standardowy
            rcall     lcd_wr

lcd_clrscr: cbi        DC_PORT,DC_PIN
            ldi        r24,0x40            ;zeruj licznik kolumn
            rcall     lcd_wr
            ldi        r24,0x80            ;zeruj licznik wierszy
            rcall     lcd_wr

            sbi        DC_PORT,DC_PIN
            ldi        r21, 252            ;504 bajty / 2 = 252 obroty petli
LCD_CLEAR_1: clr        r24                ;petla kasowania zawartości RAMu wysw.
            rcall     lcd_wr                ;wyslij bajt 0 do RAMu
            clr        r24
            rcall     lcd_wr                ;wyslij bajt 0 do RAMu
    
```

List. 1. - cd.

```

        dec     r21           ;zmniejsz licznik
        brne   LCD_CLEAR_1
        ret

SPI_init:  sbi     DDRB, PB5       ;ustawienie MOSI jako wyjście
          sbi     DDRB, PB4       ;SS musi być wyjściem dla trybu master
          cbi     PORTB, PB7      ;ustawienie SCK na L
          sbi     DDRB, PB7       ;ustawienie SCK jako wyjście
          ldi     r24,(1<<MSTR) | (1<<SPE); ;Tryb Master, fCPU/4
          out     SPCR,r24
          in      r24,SPSR        ;czyść status SPI
          ret                    ;koniec inicjalizacji SPI

lcd_wr:   cbi     CS_PORT,CS_PIN  ;SCE = L
          out     SPDR,r24        ;załaduj bajt do rejestru danych SPI
SPI_SEND_0: sbis   SPSR,SPIF
          rjmp    SPI_SEND_0     ;zaczekaj na wysłanie bajtu
          in      r24,SPDR        ;porzuć odebrany bajt
          sbi     CS_PORT,CS_PIN  ;SCE = H
          ret

```

- wysłać 504 bajty o wartości 0x00, co spowoduje wyzerowanie pamięci RAM kontrolera,
- zakończyć inicjalizację poprzez ustawienie linii SCE w stan wysoki.

Oczywiście można również ustawiać linię SCE w stan niski przed wysłaniem każdego jednego bajtu i przełączać w stan wysoki po jego wysłaniu, co zmniejsza możliwość rozsynchronizowania się transmisji, a co za tym idzie zapisania błędnych informacji do rejestrów kontrolera. Na **list. 1** przedstawiono przykładową, napisaną w assemblerze, sekwencję inicjującą interfejs SPI mikrokontrolera AVR oraz podłączony do niego wyświetlacz.

Romuald Biały