

Obsługa graficznych wyświetlaczy TFT z chipsetem FSA506

W dobie ekspansywnego rozwoju elektroniki użytkowej nikogo nie dziwi już fakt, iż nawet proste urządzenia elektroniczne wyposażane są w coraz to lepszej jakości wyświetlacze kolorowe wykonane w technologii TFT lub OLED. Spadek ich cen powoduje, że po tego typu wyświetlacze sięgają nie tylko wytwórcy sprzętu OEM, ale są one stosowane również w rozwiązaniach amatorskich. Jest to tym łatwiejsze, że producenci wyświetlaczy konstruuje coraz to nowe chipsety, które z punktu widzenia systemów mikroprocesorowych znacznie upraszczają wyświetlanie obrazu.

Przykładem układu scalonego będącego zaawansowanym sterownikiem ekranów TFT jest FSA506. Można go stosować w systemach mikroprocesorowych o różnej architekturze. Schemat funkcjonalny układu umieszczono na **rys. 1**.

W oparciu o ten układ firma Winstar zbudowała kompletny wyświetlacz TFT o oznaczeniu WF35DTIBCDB, rozdzielczości 320×240 pikseli, wyświetlający 64 kolorów. Można go kupić w różnych wersjach, różniących się liczbą linii magistrali danych, jak również wyświetlaną liczbą kolorów. Co więcej, chipset FSA506 można znaleźć również w innych panelach TFT produkowanych przez niezależnych producentów. Aby przybliżyć nieco ten ciekawy element przedstawię pokrótce jego najważniejsze cechy:

- wbudowana, wewnętrzna pamięć obrazu z 18-bitową adresacją, pozwalająca na obsługę wyświetlacza TFT o organizacji 640×240 pikseli i 262144 kolorów (18-bitów),
- wbudowane, sprzętowe funkcje rotacji wyświetlanego obrazu oraz jego przesuwania w poziomie i pionie,
- wbudowany mechanizm definiujący aktywny obszar ekranu do operacji zapisu z automatyczną inkrementacją adresów, upraszczający adresowanie pamięci obrazu,
- możliwość pracy z magistralą danych o szerokości 8, 9, 16 lub 18 bitów,
- liczba wyświetlanych kolorów 65536 lub 262144 (definiowane sprzętowo),
- wbudowany sterownik modułu podświetlenia,

– napięcie zasilania 3,3 V.

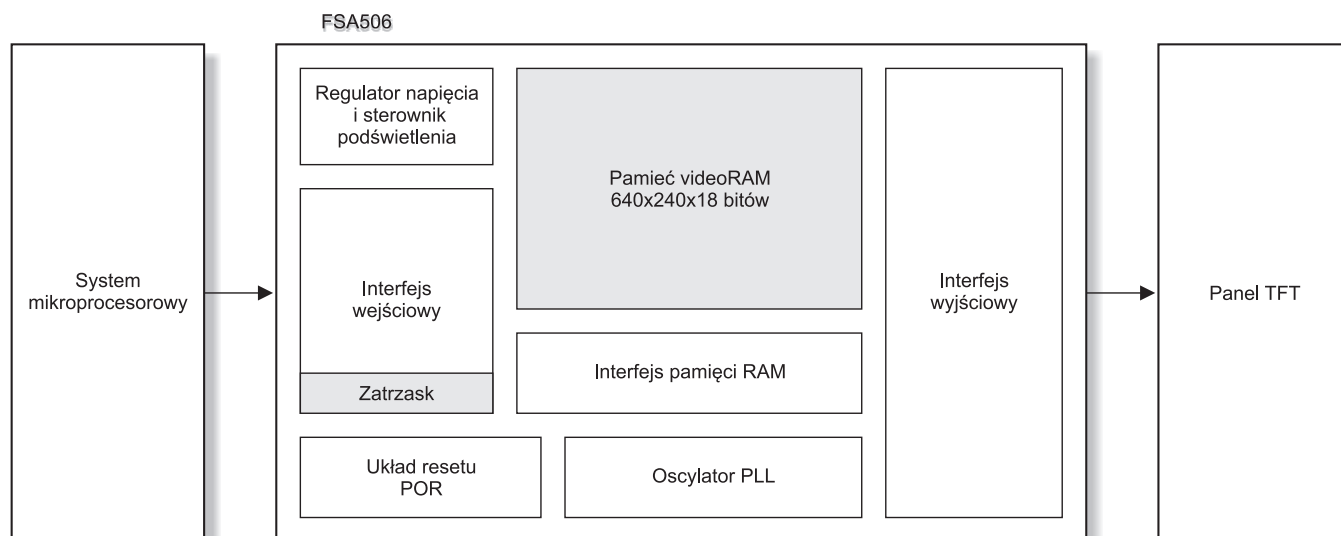
Chipset charakteryzuje się prostotą obsługi. Jak wspomniano, układ FSA506 wspiera magistrale danych o różnych szerokościach (ustawiane sprzętowo przez dedykowane wyprowadzenia chipsetu), a konkretne rozwiązanie zależy od producenta wyświetlacza TFT. W przypadku omawianego typu (WF35DTIBCDB firmy Winstar) jest to 8 bitów.

Opisywany wyświetlacz jest produkowany w wersjach z panelem dotykowym lub bez niego. Zastosowanie panelu dodatkowo podnosi jego atrakcyjność z punktu widzenia potencjalnych aplikacji.

Wyświetlacz WF35DTIBCDB wyposażono w 20-stykowe złącze ZIF z rastrem 1 mm, którego rozkład wyprowadzeń wraz z opisem funkcjonalnym umieszczono w **tab. 1**.

Przykładową aplikację układową z wykorzystaniem wyświetlacza WF35DTIBCDB zintegrowanego z panelem dotykowym oraz mikrokontrolerem ATmega32L pokazano na **rys. 2**.

Wybór mikrokontrolera nie jest krytyczny, lecz należy wybrać mikrokontroler taktowany częstotliwością rzędu 20 MHz, a to z uwagi na dużą liczbę informacji niezbędnych do przesłania zwłaszcza, jeśli



Rys. 1. Schemat blokowy układu FSA506

ne stanem wyprowadzenia RS – wysłanie polecenia sterującego oraz wysłanie bajta danych. Na listingach w dalszej części artykułu umieszczono procedury obsługujące zapis do wyświetlacza (związane z rys. 2). W treści procedur użyto aliasów dla właściwych nazw portów, co ułatwia zrozumienie treści programu. Przyjęto także, iż port sterujący sygnałem RD (sygnał żądania operacji odczytu) pozostaje zawsze w stanie wysokim, gdyż interesuje nas wyłącznie operacja zapisu.

Program nieprzypadkowo napisano w Bascom Basic. Pomijając fakt, że można go wykorzystać w aplikacjach rozwijanych z wykorzystaniem Bascoma, moim zdaniem każdy język o strukturze podobnej do Basica czy też Pascala jest na tyle prosty i czytelny, że można łatwo na jego podstawie napisać program w dowolnym języku programowania. Przy implementacji procedur sterujących należy ustawić odpowiednie kierunki portów sterujących pracą wyświetlacza. W tym przypadku, wszystkie porty ustawiane są jako wyjściowe ze stanem wysokim po włączeniu zasilania. Definicję aliasów dla portów sterujących umieszczono na list. 1, na list. 2 procedurę wysyłającą polecenie, natomiast na list. 3 procedurę wysyłającą bajt danych.

Rejestry FSA506 i ich przeznaczenie.

W tab. 2 wymieniono rejestry, dzięki którym można kontrolować sposób funkcjonowania wyświetlacza i zapis do pamięci videoRAM. Co ważne, w przypadku niektórych rejestrów konfiguracyjnych producent zaleca wpisanie do nich wartości domyślnych, dostępnych w dokumentacji. Tak naprawdę, to nie mamy innego wyjścia, gdyż dokumentacja w tym zakresie nie zagłębia się w wyjaśnianie szczegółów.

List. 1. Definicja aliasów używanych w programie

```
Data_port Alias Portc
Wr_pin Alias Portd.5
Rd_pin Alias Portd.4
Cs_pin Alias Portd.2
Rs_pin Alias Portd.6
Reset_pin Alias Portd.3
```

List. 2. Sposób wysłania słowa kontrolnego do wyświetlacza

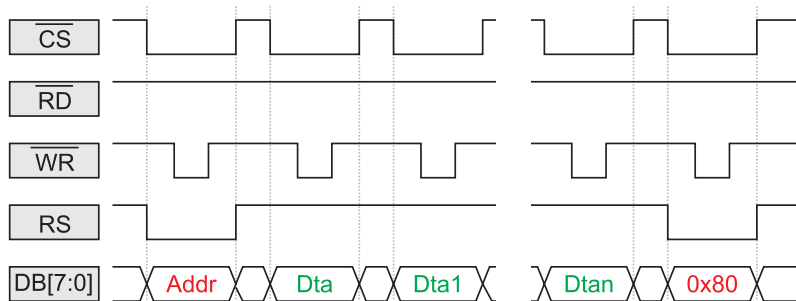
```
Sub Send_as_cmnd(byval Addr As Byte)
  Reset Rs_pin
  Reset Cs_pin
  Reset Wr_pin
  Data_port = Addr
  Set Wr_pin
  Set Rs_pin
  Set Cs_pin
End Sub
```

List. 3. Sposób wysłania danych do wyświetlacza

```
Sub Send_as_data(byval Dta As Byte)
  Reset Cs_pin
  Reset Wr_pin
  Data_port = Dta
  Set Wr_pin
  Set Cs_pin
End Sub
```

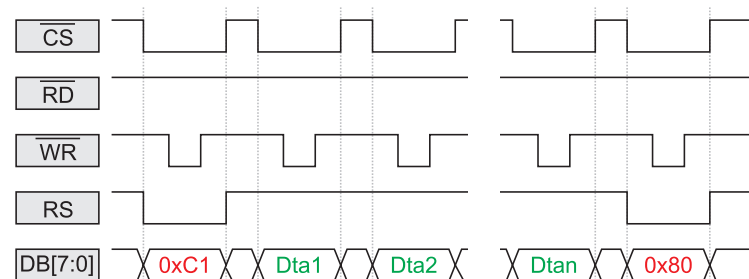
Tab.1. Rozkład wyprowadzeń wyświetlacza WF35DTIBCDB

Pin	Symbol	Opis
1	Vss	Masa zasilania
2	Vdd	Napięcie zasilania (3,3 V). Prąd zasilania ok. 300 mA.
3	BL	Sterowanie podświetleniem panelu TFT (1:włączone, 0:wylączone). Możliwość sterowania przebiegiem PWM.
4	RS	Sygnał wyboru rodzaju danych: polecenie sterujące/Dana
5	WR	Sygnał żądania operacji zapisu
6	RD	Sygnał żądania operacji odczytu
7	DB0	Magistrala danych
8	DB1	
9	DB2	
10	DB3	
11	DB4	
12	DB5	
13	DB6	
14	DB7	
15	CS	Sygnał wyboru układu (aktywacji chipsetu)
16	RST	Sygnał Reset
17, 19, 20	NC	Nie podłączone
18	FGND	Masa zasilania pamięci videoRAM



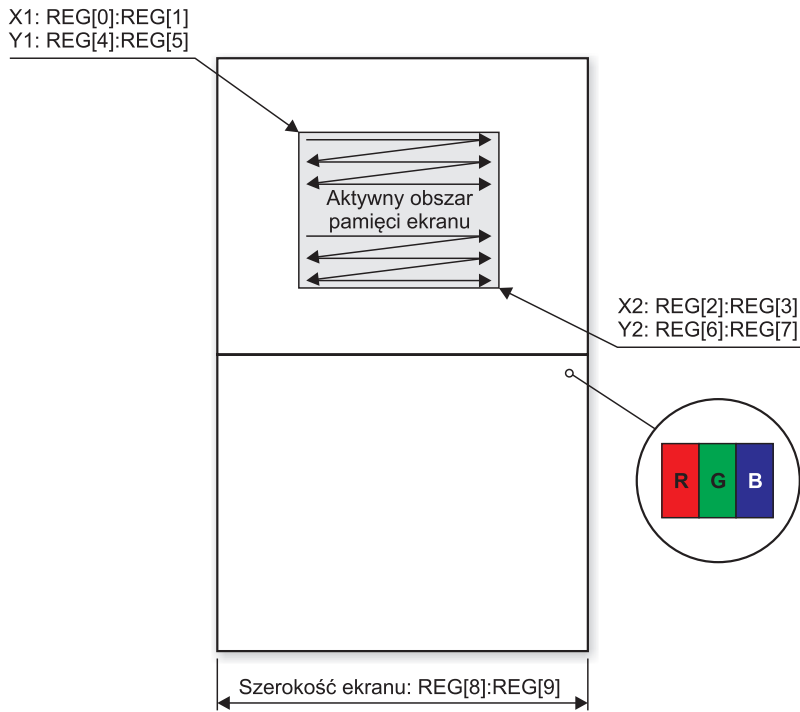
Addr – adres startowy rejestru, do którego nastąpi zapis danych
Dta – dana dla adresu startowego
Dta1 – dana dla adresu Addr + 1
Dtan – dana dla adresu Addr + n
0x80 – polecenie: koniec zapisu do rejestrów konfiguracyjnych (nie jest niezbędne)

Rys. 3. Przebiegi sygnałów sterujących zapisem do rejestrów konfiguracyjnych FSA506



0xC1 – polecenie: zapis do pamięci videoRAM
Dta1 – dana dla bieżącego adresu pamięci videoRAM. Bieżący adres pamięci videoRAM określają rejestry REG[0A]:REG[0B]:REG[0C]. Postinkrementacja adresu następuje w chwili przesłania kompletu danych przypadających na jeden piksel obrazu (w zależności od wybranego interfejsu danych i liczby kolorów będą to 1, 2 lub 3 bajty).
Dta2 – kolejna dana
Dtan – ostatnia dana
0x80 – polecenie: koniec zapisu do pamięci videoRAM

Rys. 4. Przebiegi sygnałów sterujących zapisem danych do pamięci obrazu videoRAM FSA506



Rys. 5. Wyjaśnienie działania mechanizmu autoinkrementacji

Tab. 2. Wykaz rejestrów konfiguracyjnych układu FSA506

Adres rejestru	Opis
0x00	Definiuje adres startowy aktywnego obszaru pamięci ekranu, czyli współrzędną X lewego, górnego rogu aktywnej powierzchni ekranu (bajt MSB)
0x01	Definiuje adres startowy aktywnego obszaru pamięci ekranu czyli współrzędną X lewego, górnego rogu aktywnej powierzchni ekranu (bajt LSB)
0x02	Definiuje adres końcowy aktywnego obszaru pamięci ekranu czyli współrzędną X prawego, dolnego rogu aktywnej powierzchni ekranu (bajt MSB)
0x03	Definiuje adres końcowy aktywnego obszaru pamięci ekranu czyli współrzędną X prawego, dolnego rogu aktywnej powierzchni ekranu (bajt LSB)
0x04	Definiuje adres startowy aktywnego obszaru pamięci ekranu czyli współrzędną Y lewego, górnego rogu aktywnej powierzchni ekranu (bajt MSB)
0x05	Definiuje adres startowy aktywnego obszaru pamięci ekranu czyli współrzędną Y lewego, górnego rogu aktywnej powierzchni ekranu (bajt LSB)
0x06	Definiuje adres końcowy aktywnego obszaru pamięci ekranu czyli współrzędną Y prawego, dolnego rogu aktywnej powierzchni ekranu (bajt MSB)
0x07	Definiuje adres końcowy aktywnego obszaru pamięci ekranu czyli współrzędną Y prawego, dolnego rogu aktywnej powierzchni ekranu (bajt LSB)

List. 4. Inicjalizacja wyświetlacza

```

Sub Initialize_fsa506
Dim I As Long
`Zerowanie wyświetlacza LCD
  Reset Reset_pin
  Waitms 100
  Set Reset_pin
`Ustawienia sprzętowe parametrów oscylatora PLL
  Call Send_as_cmdnd(&H40)
  Call Send_as_data(&H12)
  Call Send_as_data(&H01)
  Call Send_as_data(&H02)
`Definicja aktywnego obszaru pamięci ekranu: X=0-319, Y=0-239
  Call Send_as_cmdnd(&H00)
  Call Send_as_data(&H00)
  Call Send_as_data(&H00)
  Call Send_as_data(&H01)
  Call Send_as_data(&H3F)
  Call Send_as_data(&H00)
  Call Send_as_data(&H00)
  Call Send_as_data(&H00)
  Call Send_as_data(&Hef)
`Definicja poziomej rozdzielczości ekranu: 320 pikseli
  Call Send_as_data(&H01)
  Call Send_as_data(&H40)
`Definicja adresu startowego pamięci ekranu dla operacji zapisu: 0
  Call Send_as_data(&H00)
  Call Send_as_data(&H00)
  Call Send_as_data(&H00)
`Ustawienia rejestru konfiguracyjno-testowego
  Call Send_as_cmdnd(&H10)

```

Rejestry o adresach 0x00...0x07 pozwalają zdefiniować aktywny obszar pamięci ekranu, do którego będziemy wysyłać dane. W przypadku, gdy w danej chwili chcemy mieć dostępny cały obszar obrazu, to dla wyświetlacza o rozdzielczości 320×240 pikseli będą to odpowiednio zakresy 0...319 dla rejestrów pod adresami 0x00...0x03 i 0...239 dla 0x04...0x07. Wprowadzenie możliwości definiowania aktywnego obszaru ekranu upraszcza zapis do pamięci obrazu, gdyż kolejne dane (dla aktywnego obszaru danych) wyświetlane będą od lewej do prawej i od góry do dołu w ramach aktualnej definicji, wykorzystując mechanizm autoinkrementacji adresów pamięci videoRAM. Zwalnia to programistę z konieczności pilnowania aktualnego położenia wskaźnika bieżącego adresu. Bazując na tej właściwości możemy np. w prosty sposób wyświetlić pionową linię lub też niewielki obrazek. Wyjaśnienie tego mechanizmu zilustrowano na rys. 5.

Użyteczne procedury. W tym momencie znamy już wszystkie niezbędne informacje potrzebne do napisania własnej biblioteki obsługi wyświetlacza ze sterownikiem FSA506.

Na list. 4 zamieszczono procedurę inicjalizacją wyświetlacza. Jej wywołanie powinno nastąpić po załączeniu zasilania, przed jakimkolwiek użyciem LCD. Procedura używa mechanizmu automatycznej inkrementacji adresów rejestrów konfiguracyjnych. W związku z tym operacja zapisu wartości kolejnego rejestru konfiguracyjnego nie musi być poprzedzona wysłaniem jego numeru. Oczywiście dane muszą być przeznaczone do zapisu w kolejnych, następujących po sobie rejestrach.

Po zakończeniu pracy przez procedurę inicjalizacyjną wyświetlacz jest gotowy do pracy. Dalej przedstawię kilka użytecznych procedur związanych z obsługą podstawowych funkcji panelu. Procedury celowo nie zostały zoptymalizowane. Dzięki temu ich kod jest bardziej przejrzysty, a więc prostszy do samodzielnej analizy.

Pierwszą z nich (list. 5) jest procedura pozwalająca zdefiniować aktywny obszar pamięci obrazu dla operacji zapisu. Jej użycie upraszcza i znacznie przyspiesza wyświetlanie obrazów, których rozmiar jest mniejszy, aniżeli całkowity, fizyczny rozmiar obrazu.

Kolejną jest procedura wyświetlająca punkt na bieżącej pozycji, określonej zawartością rejestrów 0x0A...0x0C. Wskaźnik ten możemy dla przykładu ustawić za pomocą procedury *Locate_pixel()*. Procedura *Set_pixel* korzysta ze zmiennej globalnej *Crnt_colour* (typu *Word*), która przechowuje bieżący kolor obrazu. Ciało wspomnianej procedury zawiera kod *Send_as_data()*, jednak celowo unika się wywoływania tej ostatniej dla skrócenia czasu zapisu danych do pamięci ekranu. Kod procedury jest uproszczony

List. 4. c.d.

```

Call Send_as_data(&H0d)
'Ustawienia rejestru konfiguracyjnego kolejności
'składowych kolorów RGB: kolejność RGB
Call Send_as_data(&H05)
'Definicja parametrów czasowych sterownika ekranu
Call Send_as_data(&H00)
Call Send_as_data(&H00)
Call Send_as_data(&H00)
Call Send_as_data(&H10)
Call Send_as_data(&H00)
Call Send_as_data(&H44)
Call Send_as_data(&H01)
Call Send_as_data(&H40)
Call Send_as_data(&H01)
Call Send_as_data(&Hb8)
Call Send_as_data(&H00)
Call Send_as_data(&H00)
Call Send_as_data(&H00)
Call Send_as_data(&H08)
Call Send_as_data(&H00)
Call Send_as_data(&H12)
Call Send_as_data(&H00)
Call Send_as_data(&Hf0)
Call Send_as_data(&H01)
Call Send_as_data(&H09)
'Ustawienia adresu startowego dla operacji odczytu
Call Send_as_data(&H00)
Call Send_as_data(&H00)
Call Send_as_data(&H00)
'Ustawienia parametrów czasowych sterownika ekranu TFT
Call Send_as_data(&H01)
'Ustawienia rejestru sterującego: LCD włączony, obrót obrazu 0'
Call Send_as_cmnd(&H2d)
Call Send_as_data(&H08)
'Ustawienia rejestrów przesunięcia obrazu w poziomie i pionie: 0, 0
Call Send_as_cmnd(&H30)
Call Send_as_data(&H00)
Call Send_as_data(&H00)
Call Send_as_data(&H00)
Call Send_as_data(&H00)
'Ustawienia rozdzielczości ekranu w poziomie i pionie: 320, 240
Call Send_as_data(&H01)
Call Send_as_data(&H40)
Call Send_as_data(&H00)
Call Send_as_data(&Hf0)
'Koniec procedury inicjalizacyjnej wyświetlacza LCD
'Wyczyszczenie zawartości ekranu - należy wpisać 0
'pod wszystkie adresy aktywnego obszaru pamięci video RAM
'Polecenie sterujące: zapis do pamięci videoRAM
Call Send_as_cmnd(&Hc1)
For I = 1 To 153600
Reset Cs_pin
Reset Wr_pin
Data_port = 0
Set Wr_pin
Set Cs_pin
Next I
'Polecenie sterujące: koniec zapisu do pamięci videoRAM
Call Send_as_cmnd(&H80)
End Sub

```

List. 5. Ustawienie obszaru aktywnego

```

Sub Set_active_region(byval X1 As Word , Byval Y1 As Byte , Byval X2 As Word ,
Byval Y2 As Byte)
Dim Temp As Byte
'Definicja aktywnego obszaru pamięci ekranu
Call Send_as_cmnd(&H00)
Temp = High(X1)
'Adres startowy aktywnego obszaru pamięci ekranu dla osi X (MSB)
Call Send_as_data(Temp)
Temp = X1
'Adres startowy aktywnego obszaru pamięci ekranu dla osi X (LSB)
Call Send_as_data(Temp)
Temp = High(X2)
'Adres końcowy aktywnego obszaru pamięci ekranu dla osi X (MSB)
Call Send_as_data(Temp)
Temp = X2
'Adres końcowy aktywnego obszaru pamięci ekranu dla osi X (LSB)
Call Send_as_data(Temp)
'Adres startowy aktywnego obszaru pamięci ekranu dla osi Y (MSB=0)
Call Send_as_data(0)
'Adres startowy aktywnego obszaru pamięci ekranu dla osi Y (LSB)
Call Send_as_data(Y1)
'Adres końcowy aktywnego obszaru pamięci ekranu dla osi Y (MSB=0)
Call Send_as_data(0)
'Adres końcowy aktywnego obszaru pamięci ekranu dla osi Y (LSB)
Call Send_as_data(Y2)
End Sub

```

(nie zawiera poleceń sterujących) z uwagi na fakt, że jest ona przeznaczona wyłącznie do użycia w pozostałych procedurach bibliotecznych.

Kolejną procedurą jest procedura ustawiająca wskaźnik pamięci videoRAM dla operacji zapisu (list. 6).

Na list. 7 i 8 zamieszczono odpowiednio procedury służące do wyświetlenia poziomej i pionowej linii na ekranie wyświetlacza LCD. Rysując linię pionową korzysta się z mechanizmu pozwalającego określić aktywny obszar ekranu dla operacji zapisu, dzięki czemu zapis do pamięci ekranu ulega uproszczeniu.

UKŁADY INTERNETOWE

AVT966
Karta przekazników sterowana przez Internet



Dostępne wersje:
A - płytką drukowaną i dokumentacją
B - komplet elementów z płytką
C - układ zmontowany i uruchomiony

AVT953
Karta wejść z interfejsem Ethernet



Dostępne wersje:
A - płytką drukowaną i dokumentacją
B - komplet elementów z płytką
C - układ zmontowany i uruchomiony

AVT927
Uniwersalny interfejs Internetowy



Dostępne wersje:
A - płytką drukowaną i dokumentacją
B - komplet elementów z płytką
C - układ zmontowany i uruchomiony

www.sklep.avt.pl

Producent: AVT-Korporacja Sp. z o.o.
03-197 Warszawa, ul. Leszczyńska 11
tel. 022 257 84 50, fax 022 257 84 55
e-mail: handlowy@avt.pl

Adres rejestru	Opis
0x08	Definiuje poziomą rozdzielczość ekranu (bajt MSB)
0x09	Definiuje poziomą rozdzielczość ekranu (bajt LSB)

Rejestry pozwalają sterownikowi na określenie trybu pracy ekranu wybranego przez użytkownika w zakresie ustawień: portret/pejzaż. Trybu pejzaż (320×240) = wartość 320, tryb portret (240×320) = 240.

Adres rejestru	Opis
0x0A	Ustala adres startowy pamięci ekranu dla operacji zapisu (wyświetlanie obrazu). Każdy piksel obrazu (składowe RGB) reprezentowany jest przez pojedynczy adres w przestrzeni adresowej pamięci obrazu. Proces zapisu do pamięci ekranu wykorzystuje mechanizm automatycznej inkrementacji. Liczba bajtów, po wysłaniu których nastąpi automatyczna inkrementacja zależy od szerokości magistrali danych i wybranej liczby kolorów (tu 2 bajty). (bity 17...16)
0x0B	j.w. (bity 15...8)
0x0C	j.w. (bity 7...0)
0x10	Rejestr konfiguracyjno-testowy. Dla normalnej pracy sterownika ekranu wymagane jest wpisanie wartości 0x0D.
0x11	Rejestr konfiguracyjny kolejności składowych kolorów RGB dla operacji zapisu pamięci ekranu. Wartość domyślna 0x05 oznacza kolejność RGB.

Adres rejestru	Opis
0x12...0x25	Definicja parametrów czasowych sterownika ekranu TFT. Wartości domyślne odpowiednio wynoszą: (adres 0x12=) 0x00, 0x00, 0x00, 0x10, 0x00, 0x44, 0x01, 0x40, 0x01, 0xb8, 0x00, 0x00, 0x00, 0x08, 0x00, 0x12, 0x00, 0xf0, 0x01, (adres 0x25=)0x09.

Adres rejestru	Opis
0x26	Ustala adres startowy pamięci ekranu dla operacji odczytu. Każdy piksel obrazu (składowe RGB) reprezentowany jest przez pojedynczy adres w przestrzeni adresowej pamięci obrazu. (bity 17...16)
0x27	j.w. (bity 15...8)
0x28	j.w. (bity 7...0)

Adres rejestru	Opis
0x29	Rejestr ten definiuje parametry czasowe sterownika ekranu TFT. Wartość domyślna 0x01.
0x2A	Rejestr ten wykorzystywany jest wyłącznie w trybie testu. Definiuje wtedy parametry składowej R wyświetlanego obrazu testowego. Wartość domyślna 0x00.
0x2B	j.w. składowa G
0x2C	j.w. składowa B

Na list. 9 i 10 zamieszczono procedury służące do zarządzania zawartością ekranu rozumianego jako całość. Pierwsza z procedur jest przeznaczona do przesuwania obrazu w pionie i poziomie. Druga służy do obracania obrazu o predefiniowany sprzętowo kąt.

Podsumowanie

W artykule umieszczono opis podstawowych procedur związanych z obsługą modułów graficznych wyświetlaczy LCD z kontrolerem FSA506. Mam jednocześnie nadzieję, iż ten krótki kurs zachęci Czytelników do własnych eksperymentów tym bardziej, że jego cena osiągnęła poziom ceny zwykłego, graficznego, monochromatycznego wyświetlacza LCD. Zastosowanie wyświetlacza kolorowego nie tylko

Adres rejestru	Opis
0x2D	Steruje pracą wyświetlacza. Znaczenie poszczególnych bitów jest następujące: Bit 3: 0-panel wyłączony, 1-panel włączony Bity 1 i 0 – obrót obrazu: - 00 (brak obrotu), 01 (90°), 10 (270°), 11 (180°)

Adres rejestru	Opis
0x30	Definiuje przesunięcie wyświetlanego obrazu w poziomie (bity 11...8)
0x31	Definiuje przesunięcie wyświetlanego obrazu w poziomie (bity 7...0)
0x32	Definiuje przesunięcie wyświetlanego obrazu w pionie (bity 11...8)
0x33	Definiuje przesunięcie wyświetlanego obrazu w pionie (bity 7...0)

Adres rejestru	Opis
0x34	Definiuje fizyczną rozdzielczość ekranu w poziomie w odniesieniu do pamięci obrazu jak i liczby wyświetlanych kolorów (bity 11...8)
0x35	Definiuje fizyczną rozdzielczość ekranu w poziomie w odniesieniu do pamięci obrazu jak i liczby wyświetlanych kolorów (bity 7...0)
0x36	Definiuje fizyczną rozdzielczość ekranu w pionie w odniesieniu do pamięci obrazu jak i liczby wyświetlanych kolorów (bity 11...8)
0x37	Definiuje fizyczną rozdzielczość ekranu w pionie w odniesieniu do pamięci obrazu jak i liczby wyświetlanych kolorów (bity 7...0)

Wpisy do tego rejestru pozwalają sterownikowi ekranu na poprawne adresowanie pamięci obrazu. Np. wartości dla trzech konfiguracji ekranu:

- 320×240×8 bit → REG[34]=0x01, REG[35]=0x40, REG[36]=0x00, REG[37]=0xF0
- 320×480×18 bit → REG[34]=0x01, REG[35]=0x40, REG[36]=0x01, REG[37]=0xE0
- 640×240×18 bit → REG[34]=0x02, REG[35]=0x80, REG[36]=0x00, REG[37]=0xF0

Adres rejestru	Opis
0x40...0x42	Rejestry te ustalają parametry oscylatora PLL sterownika FSA506 i powinny być ustawione według domyślnych wartości dokumentacji producenta. Wartości domyślne dla kolejnych rejestrów w ramach zakresu 0x40...0x42 wynoszą: 0x12, 0x01, 0x02.

podnosi walory estetyczne, ale i użytkowe zwłaszcza jeśli weźmie się pod uwagę fakt, iż współczesne, tanie LCD są zdolne wyświetlić „jedynie” 65 tys. czy też 262 tys. kolorów... Oczywiście, ich obsługa wymaga znacznie więcej od systemu mikroprocesorowego, jednak dla współczesnych bardzo szybkich i dobrze wyposażonych mikrokontrolerów nie powinno to przedstawiać żadnego problemu. W kolejnym numerze EP zostanie zaprezentowany projekt „Cyfrowej ramki fotograficznej”, której konstrukcję oparto o dosłownie kilka niezbędnych elementów i opisywaną bibliotekę w Bascom AVT.

Robert Wołgajew, EP
robert.wolgajew@ep.com.pl

List. 6. Ustawienie wskaźnika adresu videoRAM

```

Sub Locate_pixel(byval X As Word , Byval Y As Byte)
    Dim Lpos As Long
    Dim Wpos As Word
    Dim Bpos As Byte
    'Pozycję pixela wyznaczana jako Pos=(320*Y)+X
    Lpos = Y * 320
    Lpos = Lpos + X
    'Adres startowy pamięci ekranu dla operacji zapisu
    Call Send_as_cmdnd(&H0A)
    Wpos = High(Lpos)
    Bpos = Wpos
, Bity 17 i 16
    Call Send_as_data(Bpos)
    Wpos = Lpos
    Bpos = High(Wpos)
, Bajt MSB (bity 15...8)
    Call Send_as_data(Bpos)
    Bpos = Wpos
, Bajt LSB (bity 7...0)
    Call Send_as_data(Bpos)
End Sub

```

List. 7. Rysowanie linii poziomej

```

Sub Show_pixel
    Reset Cs_pin
    Reset Wr_pin
, MSB koloru bieżącego pixela
    Data_port = High(Crnt_colour)
    Set Wr_pin
    Set Cs_pin
    Reset Cs_pin
    Reset Wr_pin
, LSB koloru bieżącego pixela
    Data_port = Crnt_colour
    Set Wr_pin
    Set Cs_pin
End Sub

```

List. 8. Rysowanie linii pionowej

```

Sub Show_hline(byval X As Word , Byval Y As Byte , Byval
Length As Word , Byval Colour As Word)
    Local Temp As Word
    'Zmiennej globalna, z której korzysta Show_pixel
    Crnt_colour = Colour
    'Ustawienie wskaźnika ekranu dla operacji zapisu
    Call Locate_pixel(X , Y)
    'Polecenie sterujące: zapis do videoRAM
    Call Send_as_cmdnd(&HC1)
    'Narysowanie linii o długości określonej w Length
    For Temp = 1 To Length
        Call Show_pixel
        Next Temp
    'Polecenie sterujące: koniec zapisu do videoRAM
    Call Send_as_cmdnd(&H80)
End Sub

```

List. 9. Przesuwanie obrazu w pionie i poziomie

```

Sub Show_vline(byval X As Word , Byval Y As Byte , Byval
Length As Byte , Byval Colour As Word)
    Local Temp As Byte
    'Zmienna globalnej, z której korzysta Show_pixel
    Crnt_colour = Colour
    'Obliczenie współrzędnej Y2 pionowej linii
    Temp = Y + Length
    Decr Temp
    'Aktywny obszar ekranu równy wielkości pionowej linii
    Call Set_active_region(X , Y , X , Temp)
    'Polecenie sterujące: zapis do pamięci videoRAM
    Call Send_as_cmdnd(&HC1)
    'Narysowanie linii o długości określonej parametrem
    Length
    For Temp = 1 To Length
        Call Show_pixel
        Next Temp
    'Polecenie sterujące: koniec zapisu do pamięci videoRAM
    Call Send_as_cmdnd(&H80)
    Call Set_active_region(0 , 0 , 319 , 239)
End Sub

```

List. 10. Obracanie obrazu o predefiniowany kąt

```

Sub Scroll_screen(byval X As Word , Byval Y As Word)
    Local Temp As Byte
    Call Send_as_cmdnd(&H30)
    Temp = High(X)
    Call Send_as_data(Temp) ,Przes. w poziomie (MSB)
    Temp = X
    Call Send_as_data(Temp) ,Przes. w poziomie (LSB)
    Temp = High(Y)
    Call Send_as_data(Temp) ,Przes. w pionie (MSB)
    Temp = Y
    Call Send_as_data(Temp) ,Przes. w pionie (LSB)
End Sub

```

TERMOMETRY I TERMOSTATY

**AVT2852 SPIRALNY TERMOMETR LED**

Zestaw pozwalający wykonać niezwykle termometr zaokrąglony. Jego oryginalność polega na sposobie wyświetlania informacji o aktualnej temperaturze. Wykorzystywane są do tego diody LED ułożone nie w tradycyjną linię, ale w spiralę. Dodatkowo tak dobrano kolory by temperatury dodatnie były obrazowane ciepłymi barwami, ujemne zaś - zimnymi.

AVT1446 Termometr z zasilaniem baterijnym

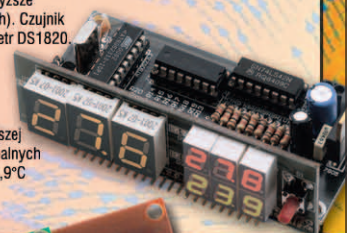
Termometr z zasilaniem baterijnym wykorzystujący do prezentacji temperatury wyświetlacz graficzny z telefonu komórkowego. Procentuje to niskim zużyciem energii i umożliwia wygodne zasilanie napięciem 3 V.

- odczyt: wyświetlacz graficzny LCD 84x48 pikseli
- odczyt temperatury cyfrowy i analogowy
- możliwość wyświetlania normalnego i inwersyjnego
- zakres pomiarowy czujnika: -55...+125°C
- zakres odczytu cyfrowego: -55...+125°C
- zakres odczytu analogowego: 0...+32°C
- rozdzielczość pomiarowa: 0,1°C
- zasilanie: 3 V (bateria)

**AVT5041 Termometr MIN-MAX**

Układ prezentuje bieżącą temperaturę, dodatkowo rejestrowane i wyświetlane są wartości najwyższe i najniższe (na osobnych polach odczytowych). Czujnik temperatury to niezawodny, scalony termometr DS1820.

- Gwarantuje on wysoką dokładność pomiaru i rozdzielczość 0,1°C.
- trzy pola odczytowe LED (trzycyfrowe)
- prezentacja aktualnej temperatury
- prezentacja temperatury najwyższej i najniższej
- kasowanie wskaźników maksymalnych i minimalnych
- zakres mierzonych temperatur -20°C...+99,9°C
- rozdzielczość 0,1°C
- zasilanie: 8...16 V

**AVT950/1 Termostat elektroniczny**

W zestawie jako czujnik temperatury zastosowano sensor półprzewodnikowy. Ustawiona temperatura jest utrzymywana z dokładnością wyznaczoną przez histerezę (określoną przez użytkownika) załączania i wyłączania przełącznika. Zastosowanie przełącznika gwarantuje pełną separację od sieci zasilającej i bezpieczeństwo osoby obsługującej termostat. Wszelkie nastawy i pomiar temperatury prezentowane są na wyświetlaczu ciekłokrystalicznym

- zakres pomiaru i regulacji temperatury -55...+99,9°C
- zakres ustawiania histerezy 0...5°C
- prezentacja temperatury nastawionej i zmierzonej
- sterowanie dołączonym odbiornikiem poprzez układ wykonawczy z przełącznikiem
- obciążalność styków przełącznika 16 A/230 V [3kV]
- komunikacja z użytkownikiem poprzez wyświetlacz LCD 1x16
- sygnalizacja stanu przełącznika: dioda LED i symbol na wyświetlaczu
- zmiana nastaw impulsatorem
- płytka o wymiarach: 104x36 mm (termostat), 34x36 mm (układ wykonawczy)
- zasilanie: 12 VDC



www.sklep.avt.pl