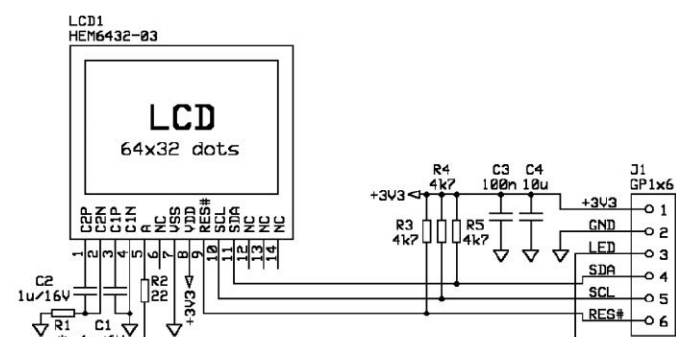


# Moduł miniaturowego wyświetlacza graficznego dla mikrokontrolera

W artykule przedstawiono oprogramowanie modułu miniaturowego wyświetlacza graficznego o wymiarach ekranu 11,2 × 5,6 mm, który może znaleźć zastosowanie w systemach mikroprocesorowych, przede wszystkim w urządzeniach noszonych (ang. wearables).



**Rysunek 1. Schemat ideowy modułu wyświetlacza graficznego 0,49"**

Dla zapoznania się z wyświetlaczem wykonałem niewielką płytkę, która ułatwiła jego dołączenie do systemu testowego. Schemat ideowy tej płytki – modułu z wyświetlaczem – pokazano na **rysunku 1**. Zastosowano na niej miniaturowy wyświetlacz graficzny typu HMC6432-03 o rozdzielczości 64×32 piksele i przekątnej 0,49". Wyświetlacz jest oferowany między innymi przez firmę Lummax pod oznaczeniem LCD-AG-064032V-VAIW W/KKK-E6 (poprzednie oznaczenie LCD-AG-V6432).

Element ten jest dość nietypowy. Na pierwszy rzut oka wyglądem zewnętrznym, budową, wymiarami oraz sposobem montażu przypomina popularny, miniaturowy wyświetlacz OLED o przekątnej ekranu 0,49". Podobieństwo do wyświetlacza OLED dotyczy także wyświetlanego obrazu – standardowo są to białe punkty na czarnym tle. Tak naprawdę jest to jednak wyświetlacz LCD, z tłem podświetlanym białą diodą LED. Wyposażono go w kontroler firmy Sitronix typu ST7567A, który komunikuje się z otoczeniem przez interfejs I<sup>2</sup>C. Wejście zasilania i wszystkie sygnały sterujące wyświetlaczem HMC6432-03 są wyprowadzone na złącze krawędziowe, umieszczone na końcu giętkiej taśmy przeznaczonej do wlutowania w obwód drukowany. Opis sygnałów występujących na złączu wyświetlacza umieszczono w **tabeli 1**.

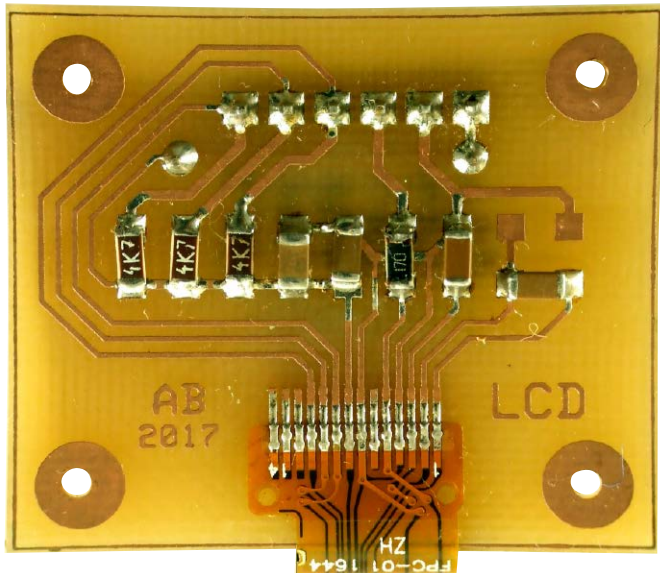
Układ aplikacyjny wyświetlacza HEM6432-03 jest nieskomplikowany. Do poprawnej pracy wyświetlacz wymaga jedynie dołączenia dwóch kondensatorów (na schemacie – C1 i C2) o pojemności od 100 nF do 1 μF, które są elementami konwerterów DC/DC wyświetlacza. Zalecane jest także filtrowanie zasilania wyświetlacza kondensatorem ceramicznym. W prezentowanym module użyto w tym

Tabela 1. Rozmieszczenie wyprowadzeń wyświetlacza HEM6432-03 i ich funkcje		
Numer wyprowadzenia	Sygnał	Opis
1	C2P (XVO)	Konwerter DC/DC
2	C2N (VO)	Konwerter DC/DC
3	C1P (VG)	Konwerter DC/DC
4	C1N (VSS)	Konwerter DC/DC
5	A	Anoda diody LED podświetlenia ekranu
6	NC	Styk nieużywany
7	VSS (K)	Masa / Katoda diody LED podświetlenia ekranu
8	VDD	Napięcie zasilania z zakresu +2,4...3,3 V
9	RES#	Reset (wejście, aktywny poziom niski)
10	SCL	Linia zegara interfejsu I <sup>2</sup> C (wejście)
11	SDA	Linia danych interfejsu I <sup>2</sup> C (typu OD)
12	NC	Styk nieużywany
13	NC	Styk nieużywany
14	NC	Styk nieużywany

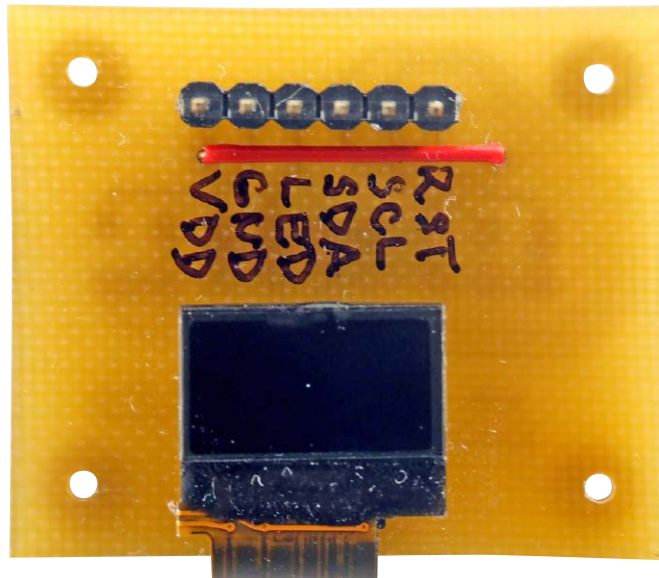
celu połączonych równolegle kondensatorów o pojemności 100 nF i 10 μF (kondensatory C3 i C4).

Wyświetlacz HEM6432-03 może być zasilany napięciem z zakresu +2,4...3,3 V, natomiast dioda LED podświetlenia ekranu napięciem +3,0 V ±0,1 V. Przy takim napięciu pobór prądu przez diodę wynosi ok. 15 mA. Aby umożliwić pracę prezentowanego modułu wyświetlacza z jednego napięcia zasilania, w obwód diody LED włączono rezystor R2 ograniczający jej prąd w wypadku zasilania napięciem wyższym od +3,0 V. Podaną na schemacie wartość rezystancji R2 dobrano dla nominalnego napięcia zasilania prezentowanego modułu, które wynosi +3,3 V.

Linie SCL i SDA wyświetlacza są przeznaczone do bezpośredniego dołączenia do wyprowadzeń mikrokontrolera, dokładnie do sprzętowego lub realizowanego w sposób programowy interfejsu I<sup>2</sup>C. Linia RES może być natomiast sterowana przez jeden z portów I/O mikrokontrolera lub też przez zewnętrzny układ generowania sygnału zerowania, jeżeli taki jest dostępny. W celu uproszczenia sposobu połączenia modułu wyświetlacza z zestawami ewaluacyjnymi mikrokontrolerów wszystkie trzy linie sygnałowe zaopatrzone w rezystory pull-up i przyłączono do napięcia zasilania modułu. Dla celów testowych wyświetlacz zamontowano na jednostronnej płytce drukowanej wykonanej metodą termotransferu. Wygląd tej płytki w widoku od spodu pokazano na **fotografii 2**, a w widoku od góry na **fotografii 3**.



Fotografia 2. Zmontowany moduł wyświetlacza LCD w widoku od spodu



Fotografia 3. Zmontowany moduł wyświetlacza LCD w widoku od góry

### Format transmisji

Podstawowym problemem przy implementacji we własnym projekcie wyświetlacza HEM6432-03 jest brak dostępności pełnej dokumentacji technicznej tego elementu. W Internecie można znaleźć jedynie fragmentaryczne dane tego wyświetlacza. Niestety, nie jest też dostępna dokumentacja kontrolera ST7567A. Co prawda można znaleźć karty katalogowe starszej wersji tego układu, tj. ST7567, jednak kontroler LCD w tej wersji nie obsługuje interfejsu I<sup>2</sup>C, który został użyty w wyświetlaczu HEM6432-03. Na szczęście analiza kart katalogowych innych sterowników monochromatycznych wyświetlaczy LCD produkowanych przez firmę Sintronix pokazuje, że producent ten we wszystkich swoich układach implementuje w interfejsie I<sup>2</sup>C ten sam protokół komunikacji. Pokazano go na rysunku 4.

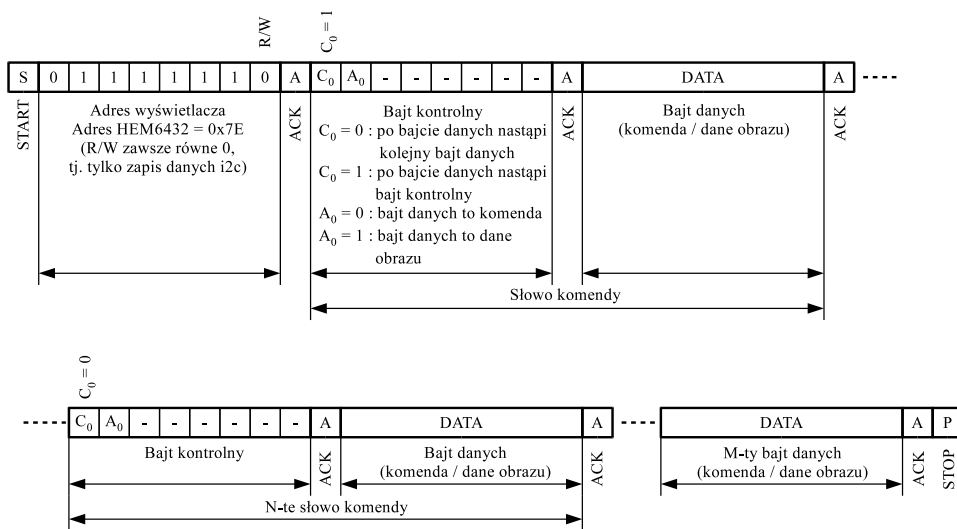
Wyświetlacz HEM6432-03 jest skonfigurowany do pracy na szynie I<sup>2</sup>C w funkcji układu slave o adresie 0x7E. Maksymalna, dopuszczalna częstotliwość zegara szyny I<sup>2</sup>C jest równa 400 kHz. W trybie obsługi interfejsu i2c sterownik ST7567A obsługuje tylko zapis komend i danych. Nie ma możliwości odczytu żadnej informacji ze sterownika przez interfejs I<sup>2</sup>C. Format komunikacji mikrokontrolera ze sterownikiem LCD różni się od typowej wymiany danych po szynie I<sup>2</sup>C, która polega zazwyczaj na zapisie/odczyście szeregu rejestrów układu slave. W wypadku sterownika ST7567A po przesłaniu adresu układu transmitowane są słowa komend. Każde z nich składa się z dwóch bajtów: bajtu kontrolnego i bajtu danych. Najstarszy bit w bajcie kontrolnym

(oznaczony C<sub>0</sub>) informuje, czy po bajcie danych będzie przesyłany bajt kontrolny kolejnego słowa komendy, czy też kolejny bajt danych. Szósty bit bajtu kontrolnego (oznaczony A<sub>0</sub>) określa natomiast, czy bajt danych zawiera komendę, czy też dane obrazu. Po bajcie kontrolnym z wyzerowanym bitem kontynuacji C<sub>0</sub> mogą być przesyłane tylko bajty danych, interpretowane przez sterownik albo komendy, jeśli bit A<sub>0</sub> w ostatnim bajcie kontrolnym był zerem, albo dane obrazu, jeśli bit A<sub>0</sub> był jedynką. Zmiana typu danych wymaga w tym przypadku wygenerowania na szynie I<sup>2</sup>C sekwencji STOP, tj. zakończenia aktualnej transmisji i rozpoczęcia nowej.

Jak widać, ten format komunikacji odbiega od najczęściej spotykanego sposobu polegającego na zapisie rejestrów układu slave. Ponieważ do takiego właśnie formatu wymiany danych są zazwyczaj przystosowane biblioteki obsługi interfejsu I<sup>2</sup>C w mikrokontrolerach, najwygodniej jest potraktować bajt kontrolny jako adres rejestru i przyjąć, że sterownik ST7567A ma dwa rejestry I<sup>2</sup>C o adresach 0x00 i 0x40. Adresy te odpowiadają bajtom kontrolnym z wyzerowanym bitem kontynuacji C<sub>0</sub> i wyzerowanym oraz ustawionym bitem A<sub>0</sub>. Dane zapisywane do pierwszego rejestru (tj. rejestru o adresie 0x00) będą interpretowane przez kontroler ST7567A jako komendy, natomiast dane zapisywane do drugiego rejestru (tj. rejestru o adresie 0x40) będą interpretowane jako dane obrazu. Takie podejście powoduje jednak, że w ramach jednej transmisji I<sup>2</sup>C będą mogły być przesyłane do kontrolera LCD tylko komendy lub tylko dane obrazu. Przesłanie

komendy i danych obrazu tym sposobem będzie wymagało dwóch transferów. Taka metoda transmisji jest jednak i tak szybsza, ponieważ w przeciwnym wypadku konieczne byłoby poprzedzanie każdej komendy/danej obrazu bajtem kontrolnym, co zwiększyłoby prawie dwukrotnie liczbę transmitowanych danych.

Przykładowe funkcje zapisu do wyświetlacza HEM6432-03 komend i danych obrazu zaprezentowano na listingu 1. Obie wysyłają dane do wyświetlacza nie bezpośrednio, lecz za pomocą funkcji niższego poziomu `i2c_WriteData()`, która dokonuje zapisu bloku danych do układu slave przyłączonego do szyny I<sup>2</sup>C. Parametry transmisji przekazywane do tej



Rysunek 4. Format transmisji danych wyświetlacza HEM6432-03

```

Listing 1. Funkcje zapisu do wyświetlacza HEM6432-03 komend i danych obrazu
void lcdHEM6432_SendCmd(uint8_t *pCommandBuf, uint8_t nByte)
{
    i2c_DataTypeDef lcdI2C;           // i2c data structure
    lcdI2C.device_addr = LCD_I2C_ADDR; // LCD i2c address (i.e. 0x7E)
    lcdI2C.data_addr = LCD_COMMAND_REG; // LCD i2c command register (i.e. 0x00)
    lcdI2C.data_size = nByte;         // number of bytes to send
    lcdI2C.pdata = pCommandBuf;       // command buffer address
    i2c_WriteData(&lcdI2C);          // send command(s) to LCD
}

void lcdHEM6432_SendData(uint8_t *pDataBuf, uint8_t nByte)
{
    i2c_DataTypeDef lcdI2C;           // i2c data structure
    lcdI2C.device_addr = LCD_I2C_ADDR; // LCD i2c address (i.e. 0x7E)
    lcdI2C.data_addr = LCD_DATA_REG;   // LCD i2c data register (i.e. 0x40)
    lcdI2C.data_size = nByte;         // number of bytes to send
    lcdI2C.pdata = pDataBuf;          // display data buffer address
    i2c_WriteData(&lcdI2C);          // send display data to LCD
}

```

```

Listing 2. Komendy i funkcja inicjalizacji wyświetlacza HEM6432-03
// HEM6432-03 LCD module full initialization data
// (define LCD_UPSIDE_DOWN if LCD module orientation is inverted)
static const uint8_t lcdInitData[] =
{
    LCD_BIAS_HIGH,                // bias select: 1/5 bias @ 1/33 duty
#ifdef LCD_UPSIDE_DOWN           // LCD module normal orientation
    LCD_SEG_DIR_NORMAL,          // segment direction: normal
    LCD_COM_DIR_REV,              // COM direction: reversed
#else                             // LCD module upside down orientation
    LCD_SEG_DIR_REV,              // segment direction: reversed
    LCD_COM_DIR_NORMAL,          // COM direction: normal
#endif
    LCD_START_LINE,               // start line number: line 0
    LCD_RR_5_5,                   // resistor regulation ratio: 5.5
    LCD_EV,                        // electronic volume level (2B command!)
    LCD_CONTRAST & LCD_EV_MSK,    // EV / contrast
    LCD_BOOSTER,                  // booster level (2B command!)
    BOOSTER_LVL_5X,               // booster level 5x
    LCD_PWR_CONTROL|PWR_VB|PWR_VR|PWR_VF, // power control: voltage booster ON,
    //                                     voltage regulator ON,
    //                                     voltage follower ON
    LCD_DISPLAY_NORMAL,           // display mode: normal
    LCD_DISPLAY_ON,               // display ON
};

void lcdHEM6432_Init(void)
{
    // I2C and DWT configuration
    // Driver assumes that i2c peripheral and DWT unit are already configured
    // by peripheral drivers and running

    // LCD_RST pin configuration
    // - enable GPIO clock
    RCC->APB2ENR |= LCD_RST_GPIO_CLK;

    // - clear GPIO CNF and MODE bit fields
    LCD_RST_GPIO->CRH &= (~(uint64_t)(GPIO_CRL_CNF0 | GPIO_CRL_MODE0) << 4*LCD_RST_PIN) >> 32);
    LCD_RST_GPIO->CRL &= (~(uint64_t)(GPIO_CRL_CNF0 | GPIO_CRL_MODE0) << 4*LCD_RST_PIN) >> 0);

    // - set LCD_RST pin as 2MHz general purpose push-pull output (i.e. conf=0x0,
    //   mode=0x2), high level
    LCD_RST_GPIO->BSRR = (1 << LCD_RST_PIN);
    LCD_RST_GPIO->CRH |= (((uint64_t)(GPIO_CRL_MODE0_1) << 4*LCD_RST_PIN) >> 32);
    LCD_RST_GPIO->CRL |= (((uint64_t)(GPIO_CRL_MODE0_1) << 4*LCD_RST_PIN) >> 0);

    // LCD hardware reset
    // LCD reset ought to be performed within 30ms after power supply voltage
    // becomes stable. Otherwise the display may operate in abnormal way. LCD
    // reset LOW pulse should last at least 1us and LCD should be reset within
    // 2us after rising edge of the reset pulse, but recommended delay times
    // should be longer than 5us and 5us respectively.
    LCD_RST_GPIO->BRR = (1 << LCD_RST_PIN); // set LCD_RST line low
    dwt_Delay(10); // wait for 10us
    LCD_RST_GPIO->BSRR = (1 << LCD_RST_PIN); // set LCD_RST line high
    dwt_Delay(10); // wait for 10us

    // LCD initialization
    // LCD controller initialization code is stored in lcdInitData[] table.
    // Initialization phase, counting from hardware reset to sending Power Control
    // command ought to be completed within 5ms!
    lcdHEM6432_SendCmd((uint8_t *)lcdInitData, sizeof(lcdInitData));
}

```

#### Wykaz elementów:

##### Rezystory: (SMD1206)

R1: niemontowany

R2: 22 W

R3, R4, R5: 4,7 kW

##### Kondensatory: (SMD1206, materiał X7R)

C1: 1 mF/6 V

C2: 1 mF/16 V

C3: 100 nF

C4: 10 mF/6 V

##### Inne:

LCD1: wyświetlacz LCD 0,49" typu HEM6432-03 lub LCD-AG-06432V-VAIW W/KKK-E6

J1: listwa gold pin 1x6, raster 2,54 mm

funkcji są opisane w strukturze *lcdI2C*, której kolejne pola zawierają: adres na magistrali I<sup>2</sup>C zapisywanego układu (pole *lcdI2C.device\_addr*), adres rejestru, do którego jest wykonywany zapis (pole *lcdI2C.data\_addr*), liczbę zapisywanych bajtów danych (pole *lcdI2C.data\_size*) oraz adres bufora z danymi do zapisu (pole *lcdI2C.pdata*).

Jak wspomniano, za pośrednictwem interfejsu I<sup>2</sup>C nie jest możliwy odczyt danych ze sterownika ST7567A. Wyświetlacz HEM6432-03 nie obsługuje więc wszystkich komend kontrolera ST7567A, a tylko te związane z operacjami zapisu. Ich listę zawiera tabela 2.

## Inicjalizacja wyświetlacza

Obsługa wyświetlacza HEM6432-03 nie różni się od programowania innych wyświetlaczy z wbudowanym układem kontrolera. Pierwszym krokiem jest zawsze skonfigurowanie modułu do pracy. Konfiguracja wyświetlacza HEM6432-03 rozpoczyna się od wygenerowania impulsu reset na linii RES#. Minimalny czas trwania impulsu reset jest równy 1 ms, aczkolwiek w opisie aplikacji kontrolera ST7567 firma Sintronix zaleca minimalny czas trwania impulsu reset równy 5 ms. Reset wyświetlacza następuje maksymalnie po czasie 2 ms od narastającego zbocza impulsu resetu, chociaż i w tym miejscu producent w opisie aplikacji kontrolera zaleca oczekiwanie przez minimum 5 ms przed podjęciem kolejnego kroku inicjalizacji wyświetlacza. Warto wspomnieć, że na tym etapie konfiguracji wyświetlacza sprzętowy reset nie może zostać zastąpiony komendą *Reset*, ponieważ programowy reset układu nie inicjalizuje wszystkich obwodów kontrolera ST7567A. Dokładny wykaz różnic pomiędzy tymi dwoma rodzajami resetu kontrolera można znaleźć w dokumentacji układu ST7567.

W kolejnym kroku konfiguracji wyświetlacza do kontrolera są wysyłane komendy inicjalizujące poszczególne obwody. Ponieważ nieznaną są ustawienia generatorów napięć dla wyświetlacza HEM6432-03, dobrano je doświadczalnie pod kątem uzyskania obrazu o jak najlepszej jakości. Zależnie od docelowego położenia modułu wyświetlacza w urządzeniu, wyświetlacz może być skonfigurowany do pracy w orientacji z taśmą przyłączeniową u dołu ekranu (fotografia 5) bądź też u góry

ekranu (fotografia 6). Orientację ekranu wybiera się za pomocą komendy *SEG Direction* ustalającej kierunek skanowania ekranu oraz *COM Direction* ustalającej kierunek skanowania elektrod COM.

Zestaw komend konfiguracyjnych wyświetlaczy dla obu orientacji ekranu (tj. normalnej i odwróconej) oraz kod funkcji inicjalizującej *lcdHEM\_Init()* przedstawiono na listingu 2. Kod ten jest przeznaczony dla mikrokontrolera STM32F107 i był testowany za pomocą zestawu uruchomieniowego STM32 Butterfly. Warto zauważyć, że funkcja *lcdHEM\_Init()* inicjalizująca wyświetlacz wykorzystuje do odmierzenia opóźnień w trakcie generowania impulsu resetu funkcję niższego poziomu *dwt\_Delay()*. Funkcja ta zapewnia odmierzenie skalibrowanych opóźnień w oparciu o odczyt stanu licznika zliczającego takty zegara jednostki centralnej. Licznik ten znajduje

się w module DWT (tj. Data Watchpoint and Trace), który jest standardowo implementowany w rdzeniach Cortex-M3/M4/M7 jako jeden z układów wspomagających debugowanie programu i monitorowanie pracy mikrokontrolera.

### **Pamięć obrazu**

Kontroler ST7567A jest w stanie obsługiwać wyświetlacz LCD o maksymalnej rozdzielczości 132×65 pikseli. W tym celu jest

on wyposażony w pamięć DDRAM liczącą 8580 bitów, której każdy bit odpowiada jednemu pikselowi. Pamięć ta jest zorganizowana w postaci 9 stron po 132 bajty. Każda strona pamięci odpowiada 8 kolejnym liniom obrazu i 132 kolumnom, natomiast każdy bajt na stronie odpowiada 8 kolejnym pikselom w kolumnie obrazu, przy czym najmłodszy bit odpowiada pikselowi w linii o najniższym numerze. Wyjątkiem jest ostatnia strona pamięci (tj. strona o adresie 8), na której zaimplementowano w słowach pamięci tylko najmłodszy

**Tabela 2. Komendy sterujące wyświetlaczem HEM6432-03**

L.p.	Komenda	Kod komendy (b7..b0)	Opis
1	Display ON/OFF	10 10 11 1 D	Włączenie / wyłączenie wyświetlacza: D=0: wyświetlacz wyłączony D=1: wyświetlacz włączony
2	Set Start Line	0 1 S5 S4 S3 S2 S1 S0	Ustawienie numeru linii, od której rozpoczyna się wyświetlanie: S5...S0: numer linii
3	Set Page Address	10 11 Y3 Y2 Y1 Y0	Ustawienie adresu strony pamięci DDRAM: Y3...Y0: adres strony pamięci
4	Set column address (MSN)	0 0 0 1 X7 X6 X5 X4	Ustawienie MSB adresu kolumny pamięci DDRAM: X7...X4: MSN adresu kolumny pamięci
5	Set Column Address (LSN)	0 0 0 0 X3 X2 X1 X0	Ustawienie LSB adresu kolumny pamięci DDRAM: X3...X0: LSN adresu kolumny pamięci
6	SEG Direction	10 10 0 0 0 MX	Ustawienie kierunku skanowania segmentów wyświetlacza: MX=0: kierunek normalny MX=1: kierunek odwrócony
7	Inverse Display	10 10 0 11 INV	Inwersja wyświetlanego obrazu: INV=0: obraz normalny (białe punkty na czarnym tle) INV=1: obraz negatywowany (czarne punkty na białym tle)
8	All Pixel ON	10 10 0 10 AP	Włączenie wszystkich pikseli obrazu: AP=0: normalna praca wyświetlacza AP=1: wszystkie piksele włączone
9	Bias Select	10 10 0 0 1 BS	Ustawienie współczynnika podziału napięcia polaryzacji LCD: BS=0: bias = 1/6 @ duty = 1/33 BS = 1: bias = 1/5 @ duty = 1/33
10	Reset	111 0 0 0 10	Programowy reset sterownika ST7567A
11	COM Direction	11 0 0 MY ---	Ustawienie kierunku skanowania elektrod COM wyświetlacza: MY=0: kierunek normalny MY=1: kierunek odwrócony (- : dowolna wartość bitu, tj. 0 albo 1)
12	Power Control	0 0 1 0 1 VB VR VF	Sterowanie obwodami zasilania wyświetlacza: VB: powielacz napięcia (0 – wyłączony, 1 – włączony) VR: stabilizator napięcia (0 – wyłączony, 1 – włączony) VF: wtórnik napięciowy (0 – wyłączony, 1 – włączony)
13	Regulation Ratio	0 0 1 0 0 RR2 RR1 RRO	Ustawienie współczynnika podziału drabinki rezystorów: RR2...RRO=000: współczynnik regulacji 3,0 RR2...RRO=001: współczynnik regulacji 3,5 RR2...RRO=010: współczynnik regulacji 4,0 RR2...RRO=011: współczynnik regulacji 4,5 RR2...RRO=100: współczynnik regulacji 5,0 RR2...RRO=101: współczynnik regulacji 5,5 RR2...RRO=110: współczynnik regulacji 6,0 RR2...RRO=111: współczynnik regulacji 6,5
14	Set EV	10 0 0 0 0 1 0 0 EV5 EV4 EV3 EV2 EV1 EV0	Ustawienie napięcia regulacji kontrastu wyświetlacza (komenda złożona z 2 bajtów !): EV5...EV0: napięcie regulacji kontrastu
15	Set Booster	111 11 0 0 0 0 0 0 0 0 0 BL	Wybór stopnia powielania napięcia (komenda złożona z 2 bajtów!): BL= 0: powielenie 4× BL= 1: powielenie 5×
16	NOP	111 0 0 0 11	Brak operacji
17	TEST	111 11 11 -	Komenda zarezerwowana dla testów, nie używać! (- : dowolna wartość bitu, tj. 0 albo 1)
18	Power Save	10 10 11 10 10 10 0 10 1	Włączenie trybu obniżonego poboru mocy – komenda złożona z dwóch komend podstawowych: Display OFF+All Pixel ON
19	Power Save OFF	10 10 0 10 0 10 10 11 11	Wyłączenie trybu obniżonego poboru mocy – komenda złożona z dwóch komend podstawowych: All Pixel ON OFF + Display ON



Fotografia 5. Wyświetlacz pracujący w orientacji normalnej, z taśmą u dołu ekranu



Fotografia 6. Wyświetlacz pracujący w orientacji odwrotnej, z taśmą u góry ekranu

bit b0. Podstawowym przeznaczeniem tej strony pamięci jest sterowanie wyświetlaniem ikon.

Rozdzielczość ekranu wyświetlacza HEM6432-03 jest znacznie mniejsza niż pamięć kontrolera ST7567A. W związku z tym tylko część pamięci DDRAM kontrolera ST7567A jest używana w opisywanym wyświetlaczu. Schemat przypisania poszczególnych pikseli wyświetlacza HEM6432-03 do komórek pamięci DDRAM kontrolera LCD przedstawiono na rysunku 7. Ponieważ używane w wyświetlaczu HEM6432-03 sterowniki kolumn kontrolera ST7567A są rozłożone niesymetrycznie, przypisanie komórek pamięci DDRAM do poszczególnych pikseli jest różne dla orientacji normalnej i odwróconej ekranu wyświetlacza. Pozostałe, niemapowane komórki pamięci DDRAM w kontrolerze ST7567A są niestety nie do użytku. Mogą one, co prawda, być zapisywane, ale ponieważ tryb obsługi interfejsu I<sup>2</sup>C w układzie ST7567A nie daje możliwości odczytu danych, nie można użyć tych komórek np. w roli dodatkowego bufora obrazu.

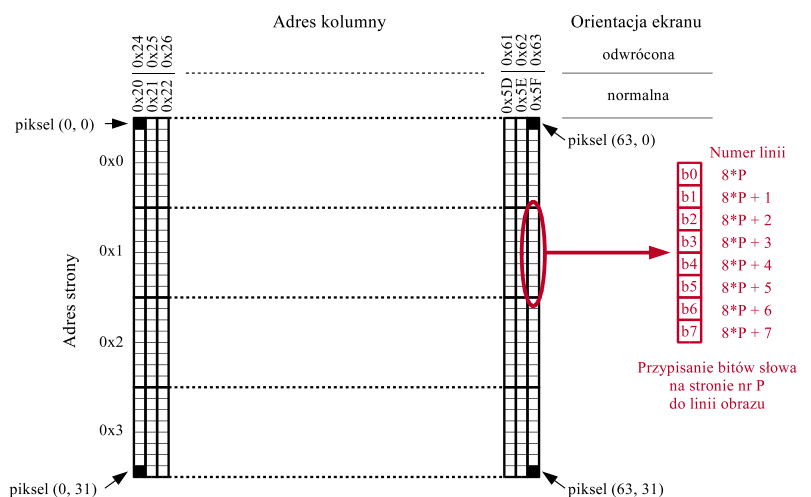
Zapis danych do pamięci DDRAM wyświetlacza HEM6432-03 rozpoczyna się od ustawienia adresu strony pamięci komendą *Set Page Address* oraz adresu kolumny komendami *Set column address (MSN)* i *Set column address (LSN)*. Proponowaną funkcję realizującą tę operację przedstawiono na listingu 3. Następnie do pamięci mogą być zapisywane dane obrazu. Po zapisie każdego bajtu danych adres kolumny jest automatycznie inkrementowany, aż do osiągnięcia końca przestrzeni adresowej na danej stronie pamięci (tj. wartości 131), po czym licznik adresujący kolumny zatrzymuje się. Adres strony pamięci nie jest automatycznie inkrementowany. Zmiana adresu strony pamięci wymaga wysłania do wyświetlacza komendy *Set Page Address* oraz ustawienia licznika adresującego kolumny na początku strony komendami *Set column address (MSN)* i *Set column address (LSN)*.

Jak można zauważyć, organizacja pamięci kontrolera ST7567A jest przystosowana do wyświetlania tekstów. Każda strona pamięci obrazu zawiera jeden

```
Listing 3. Funkcja ustawienia wskaźnika zapisu na wybranej pozycji w pamięci obrazu
#define LCD_X_SIZE          64          // LCD horizontal size (i.e. 64 columns)
#define LCD_Y_SIZE          32          // LCD vertical size (i.e. 32 lines)
#define LCD_PAGE_SIZE       8           // LCD DDRAM page size (i.e. 8 lines)
#ifdef LCD_UPSIDE_DOWN
#define LCD_COL_OFFSET      0x20       // LCD DDRAM column address offset
#else
#define LCD_COL_OFFSET      0x24       // LCD DDRAM column address offset
#endif

void lcdHEM6432_GoToXY(uint8_t Xpos, uint8_t Ypos)
{
    uint8_t lcdPosBuf[3];              // cursor position buffer
    // Check x coordinate limit (i.e. 0 <= x <= 63)
    if (Xpos >= LCD_X_SIZE) Xpos = LCD_X_SIZE-1;
    // Convert Xpos to column number
    Xpos += LCD_COL_OFFSET;
    // Store ,Set column MSN address' command code with MSN column number
    lcdPosBuf[0] = LCD_COL_ADDR_MSN | (Xpos >> 4);
    // Store ,Set column LSN address' command code with LSN column number
    lcdPosBuf[1] = LCD_COL_ADDR_LSN | (Xpos & 0x0F);
    // Check y coordinate limit (i.e. 0 <= y <= 3)
    if (Ypos >= LCD_Y_SIZE/LCD_PAGE_SIZE) Ypos = LCD_Y_SIZE/LCD_PAGE_SIZE-1;
    // Store ,Set page address' command code with page number
    lcdPosBuf[2] = LCD_PAGE_ADDR | Ypos;
    // Send command to LCD
    lcdHEM6432_SendCmd(lcdPosBuf, 3);
}
```

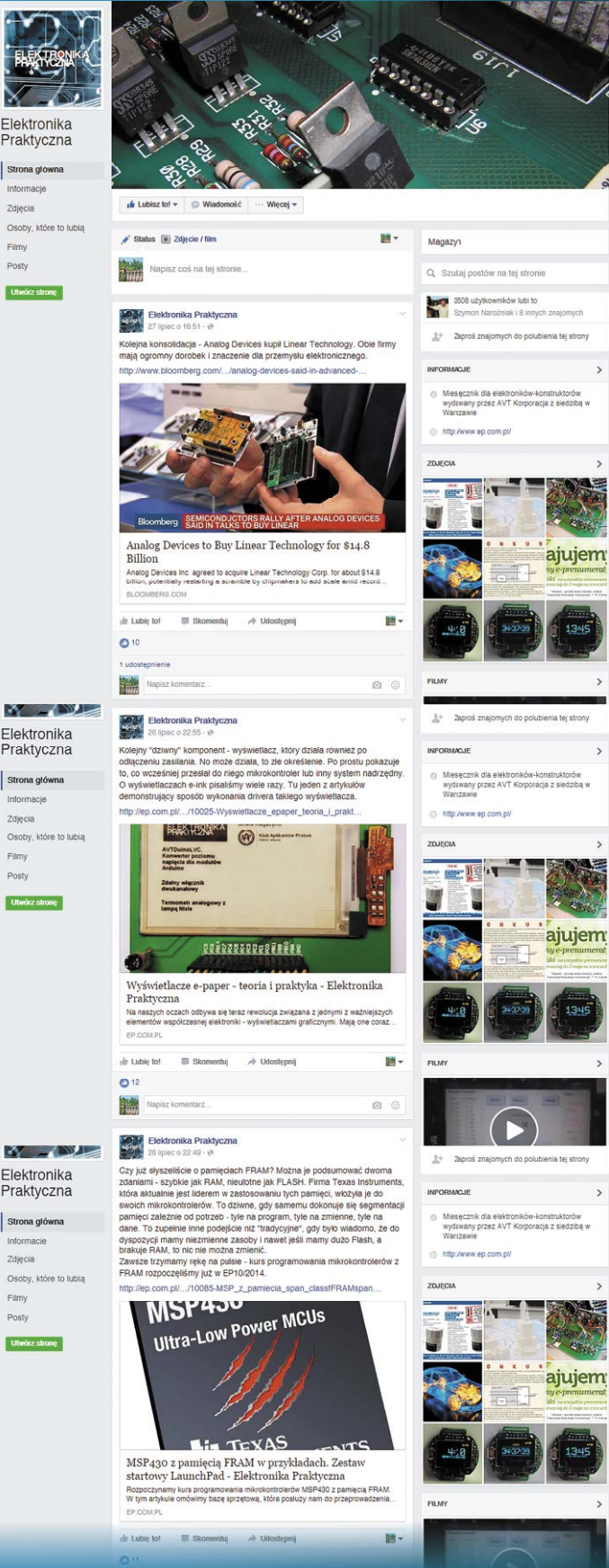
wiersz tekstu wyświetlanego standardową czcionką o rozmiarze 5×7 pikseli. Wyświetlenie znaku sprowadza się tutaj do skopiowania do pamięci DDRAM wyświetlacza kształtu znaku ze zdefiniowanej w programie tablicy znaków. Proponowaną funkcję



Rysunek 7. Mapowanie pamięci DDRAM do pikseli wyświetlacza HEM6432-03

# ELEKTRONIKA PRAKTYCZNA

## na facebook



```
Listing 4. Funkcja wyświetlania pojedynczego znaku na aktualnie wybranej pozycji kursora
void lcdHEM6432_PutChar(uint8_t char_code)
{
    uint8_t i;
    uint8_t lcdCharPatBuf[6];           // loop counter
    // character pattern buffer
    // Check character code and convert it to the printable one
    if ((char_code < , ) || (char_code > , ~')) char_code = , ;
    // Convert character code to char patterns ROM table index
    char_code -= , ;
    // Copy character pattern to buffer
    for (i = 0; i < 5; i++)
    {
        lcdCharPatBuf[i] = LCDFontROM[char_code][i];
    }
    // Add one pixel space between characters
    lcdCharPatBuf[5] = 0x00;
    // Send character pattern to LCD
    lcdHEM6432_SendData(lcdCharPatBuf, 6);
}
```



Fotografia 8. Wyświetlacz HEM6432-03 z włączoną inwersją obrazu

realizującą tę operację przedstawiono na listingu 4. Jak łatwo policzyć, wyświetlacz HEM6432-03 może zobrazować na ekranie 4 linie tekstu liczące po 10 znaków o wymiarach 5x7 pikseli.

Zaimplementowany w kontrolerze ST7567A sposób organizacji pamięci obrazu w połączeniu z brakiem możliwości odczytu danych z tej pamięci przez interfejs I<sup>2</sup>C powodują, że tworzenie grafiki bezpośrednio na ekranie wyświetlacza HEM6432-03 jest w zasadzie niemożliwe. Najprościej jest w tym celu używać lokalnego bufora obrazu, na którym przeprowadzane są wszystkie operacje graficzne, a którego zawartość jest następnie przepisywana do pamięci DDRAM wyświetlacza. Ewentualnie można poprzestać na ładowaniu do pamięci wyświetlacza predefiniowanych grafik, przechowywanych w programie w postaci tablic.

Dysponując przedstawionymi powyżej informacjami, można bez większego trudu napisać bardziej zaawansowane funkcje wyświetlające na wyświetlaczu HEM6432-03 teksty czy też predefiniowane grafiki. Dołączony do artykułu kod demonstracyjny programu zawiera w module `lcdHEM6432.c` kompletną bibliotekę obsługi omawianego wyświetlacza, wspierającą również oferowane przez ten wyświetlacz sprzętowe tryby obsługi obrazu, np. inwersję (fotografia 8).

Aleksander Borysiuk  
alex\_priv@wp.pl

Bibliografia:

- HEM6432-03 LCD Display, Karta katalogowa układu
- ST7567. 65x132 Dot Matrix LCD Controller/Driver, Karta katalogowa układu, Sitronix

REKLAMA