

MSP430: mikrokontrolery, które (prawie) nie pobierają prądu, część 5

W urządzeniach mikroprocesorowych napotyka my wyświetlacze wykonane w różnych technologiach, w tym wyświetlacze ciekłokrystaliczne. Te z kolei występują w całym zakresie rozmiarów i wykonania, zdolne są wyświetlać wielokolorową grafikę lub tylko monochromatyczny tekst. Mogą być wyposażone w podświetlenie, które daje im niezaprzeczalną zaletę – czytelność w niemal każdych warunkach oświetleniowych. Są ładne, kolorowe, zużywają masę energii i... zazwyczaj nie pasują do mikrokontrolerów MSP430. Aby zapewnić jednocześnie czytelny interfejs użytkownika i długi czas życia urządzenia, liczony w miesiącach i latach, moduł wyświetlacza musi zostać zastąpiony przez wyświetlacz ciekłokrystaliczny sterowany przez wbudowany w mikrokontroler sterownik.

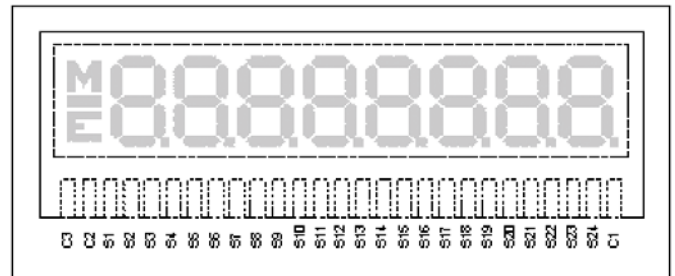
Używając określenia „wyświetlacz ciekłokrystaliczny” mamy najczęściej na myśli te mocno rozbudowane moduły. Z kolei słowo „panel” coraz częściej kojarzone jest z dużym kolorowym wyświetlaczem wyposażonym w ekran dotykowy. Jednak w świecie MSP430 te urządzenia są rzadkością a „panelem LCD” zwykło się nazywać podstawowy element każdego wyświetlacza – szklaną płytkę z zamkniętą wewnątrz warstwą ciekłego kryształu. To właśnie do pracy z takim „surowym” wyświet-

laczem przeznaczony jest sterownik, jaki znajdziemy w procesorach MSP430. Niestety, panel LCD to produkt daleki od standardu. Projektowany jest zazwyczaj do zastosowania w danym typie urządzenia i do niego dostosowana jest treść i forma graficzna znaków. Utrudnia to znacznie użycie we własnym projekcie panelu z demontażu, bo rzadko bywa uniwersalny. Poza nielicznymi wyjątkami, jak się wkrótce okaże.

Znajdujemy rozwiązanie

Można powiedzieć, że wygodne w użyciu panele LCD to rzecz szalenie popularna – wyświetlacz ciekłokrystaliczny, który dziś omawiamy, jest bowiem przeznaczony do kalkulatora elektronicznego. Ważne, by był to najprostszy

Interesującym elementem wyposażenia niektórych mikrokontrolerów z rodziny MSP430 jest sterownik wyświetlacza ciekłokrystalicznego. W tej części kursu dowiemy się jak go użyć. Dowiemy się również, jak podłączyć do mikrokontrolera wyświetlacz LCD.



Rys. 13. Rozmieszczenie wyprowadzeń wyświetlacza zastosowanego w prezentowanym projekcie

model czterodziałaniowy z 8-pozycyjnym wyświetlaniem. Jest bardzo prawdopodobne, że wymontowany z takiego kalkulatora panel okaże się wyświetlaczem trójpodłożowym o 27 wy-

prowadzeniach, z których 3 to wyprowadzenia sterujące podłożami, a pozostałe 24 to wyprowadzenia segmentów. Bardzo prawdopodobne, że rozkład tych wyprowadzeń będzie przy-

Wyświetlacz LCD w skrócie

Wiemy, że zachowanie ciekłego kryształu jest kontrolowane przez pole elektryczne. Aby wyświetlacz „świecił” trzeba selektywnie pobudzać warstwę ciekłych kryształów napięciem elektrycznym przyłożonym do jego wyprowadzeń. Zazwyczaj na płytkach szklanych wyświetlacza nałożone są w procesie metalizacji pola o kształcie wyświetlanych znaków połączonych ścieżkami metalicznymi z zewnętrznymi wyprowadzeniami. Aby wyjaśnić teraz sposób sterowania wyprowadzeń posłużę się analogią do siedmiosegmentowych wyświetlaczy LED.

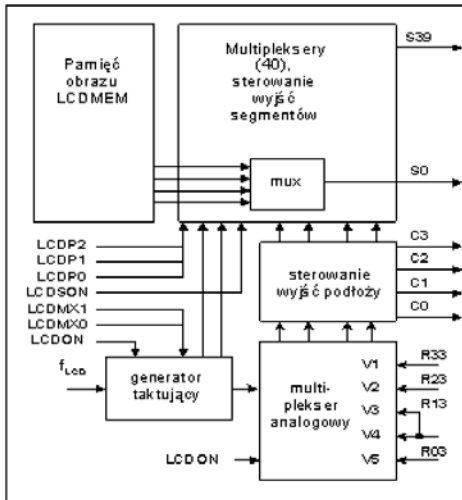
Pojedynczy, 7-segmentowy wyświetlacz LED ma 9 wyprowadzeń elektrycznych. Jedno z nich to elektroda wspólna dla wszystkich siedmiu segmentów (np. wspólna anoda), pozostałe 8 to elektrody zasilające segmenty.

Włączenie segmentu takiego wyświetlacza wymaga podania napięcia dodatniego na elektrodę wspólną, a ujemnego na jedną z pozostałych. Podłączenie wielu katod do ujemnego napięcia spowoduje jednoczesne świecenie wielu segmentów. W taki sam sposób jest zbudowany wyświetlacz ciekłokrystaliczny jednopodłożowy – wszystkie segmenty świecące włączone są pomiędzy wyprowadzenie wspólne, zwane podłożem (będziemy je określać C od angielskiego *common*), a jedno z wyprowadzeń segmentów (S).

Wadą takiego rozwiązania jest duża liczba wyprowadzeń potrzebna do sterowania segmentów. W technice LED w wielocyfrowych wyświetlaczach stosujemy dla uproszczenia sterowanie multipleksowe, w przypadku wyświetlaczy ciekłokrystalicznych również. Sterowanie jest analogiczne, wyświetlacz posiada wiele elektrod typu C i wiele elektrod typu S. Mikrokontrolery MSP430 mogą współpracować z wyświetlaczami o liczbie podłoży od 1 do 4.

Wydawać by się mogło, że tak proste sterowanie można wykonać bez specjalizowanego sterownika LCD. Tak jednak nie jest: od strony napięciowej, sterowanie LCD jest znacznie bardziej skomplikowane niż LED. Tylko jednopodłożowy wyświetlacz LCD steruje się w sposób zbliżony do LED. Wielopodłożowe wyświetlacze ciekłokrystaliczne wymagają przebiegów sterowania w postaci sekwencji napięć o skomplikowanym kształcie schodkowym.

Czytelników zainteresowanych poznaniem szczegółów zachęcamy do zakupu Elektroniki Praktycznej Plus „Displays” (<http://www.avt.pl/pisma.php?id=41>).



Rys. 14. Budowa kontrolera wyświetlacza LCD w mikrokontrolerze MSP430F449

pominał ten przedstawiony na rys. 13.

Sterowanie

Opis sterowania powinien rozpocząć się wyjaśnieniem parametrów charakterystycznych wyświetlacza. Do takich parametrów należą:

- U_{nom} – nominalne napięcie pracy,
- BIAS – maksymalne napięcie polaryzacji nie powodujące włączenia segmentu,
- MUX – liczba podłoży,
- Viewing Direction (kierunek patrzenia) – kierunek, z którego wyświetlane symbole są najbardziej czytelne,
- F_{frame} – częstość odświeżania.

„Nasz” wyświetlacz ma parametry: $U_{nom}=2,95$ V, $BIAS=1/2$, MUX3, 6 Hz, 64 Hz.

Sterownik LCD wbudowany w mikrokontrolery MSP430 może współpracować z wyświetlaczem jednopodłożowym oraz wyświetlaczem wielopodłożowym o parametrach: MUX2, BIAS1/2; MUX3, BIAS1/3 oraz MUX4, BIAS1/3. Sterownik LCD, zastosowany w mikrokontrolerach F41x/F43x/F44x ma regulowaną częstość odświeżania, a moduł LCD_A (zastosowany w serii F42x0), może również

regulować kontrast.

Na rys. 14 pokazano schemat blokowy sterownika LCD. Umieszczone na rysunku oznaczenia sygnałów mają swoje odpowiedniki w plikach nagłówkowych. Będziemy się nimi posługiwać podczas objaśniania jego pracy.

G ł ó w n e

elementy sterownika LCD to pamięć obrazu, generator taktujący i matryca multiplekserów, przygotowująca sygnały wyprowadzeń segmentów o podłożu oraz multiplekser analogowy, który w połączeniu z zewnętrznym dzielnikiem napięcia stanowi generator napięć sterujących. Do rezystorów dzielnika napięciowego można równolegle dołączyć niewielkie pojemności. Wpływa to pozytywnie na kształt przebiegów sterujących i w wielu przypadkach zwiększa kontrast i czytelność.

Pamięć obrazu procesora F44x obejmuje 20 komórek, z których każda może znaleźć swój odpowiednik na polu LCD w postaci świecącego obszaru. Każdy bajt pamięci LCDMEM reprezentuje osiem elementów świecących, połączonych z dwoma sąsiednimi wyprowadzeniami S i wszystkimi czterema podłożami. Na rys. 15 przedstawiono mapę pamięci, na której zaznaczono położenie segmentów należących do pierwszych dwóch cyfr opisywanego wyświetlacza.

Generator napięć służy do wytworzenia dwóch lub więcej napięć, które poprzez multiplekser są podawane na LCD. W przypadku wyświetlacza

jednopodłożowego (statycznego) napięcia są dwa (zazwyczaj 0 V i Vcc). Wtedy na świecących segmentach pojawia się napięcie prostokątne o amplitudzie Vcc, gdy mają one świecić lub 0 V, gdy pozostają wygaszone. Wyświetlacze wielopodłożowe wymagają do pracy większej liczby napięć, których sekwencja decyduje o zachowaniu segmentów. Minimalną liczbę napięć określa parametr BIAS – dla $BIAS=1/2$ wymagane są dwa potencjały o wartość $1/2 U_{nom}$ i U_{nom} oraz potencjał 0 V. W przypadku naszego wyświetlacza, którego $BIAS=1/2$ zastosujemy tryb $BIAS=1/3$, gdyż takim dysponuje MSP430. Generator napięć operuje wtedy napięciami 0 V, $1/2 V_{cc}$, $2/3 V_{cc}$ oraz Vcc. Na kontaktach podłoży (sygnały COM0...COM3) i segmentów (sygnały S0 i dalsze) pojawiają się przebiegi schodkowe.

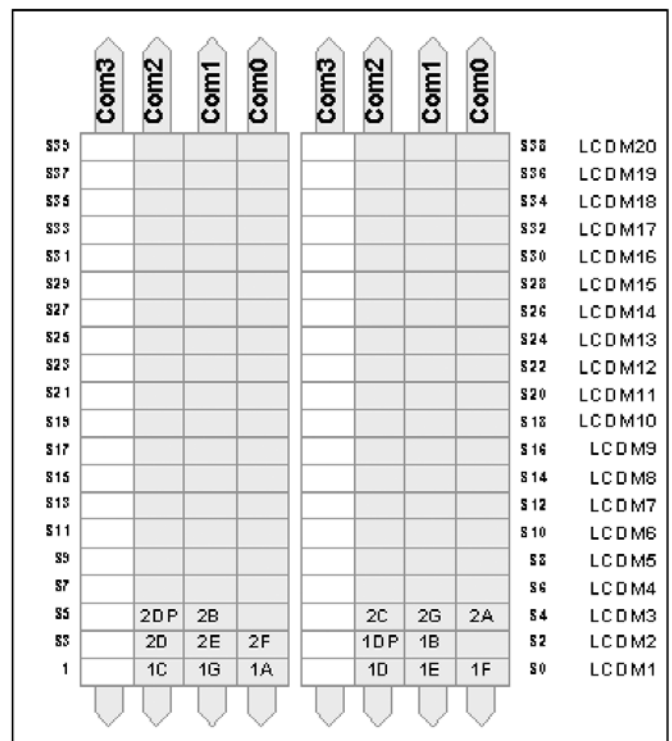
Na rys. 16 widzimy przykład sterowania dwóch segmentów wyświetlacza włączonych pomiędzy sy-

gnały C0 oraz S1 i S2. Dwa ostatnie wykresy obrazują przebieg napięcia na segmencie włączonym pomiędzy C0 i S1, który pozostaje wygaszony (napięcie różnicowe nie przekracza $1/3V_{cc}$, czyli wymagane BIAS jest spełnione) oraz na segmencie świecącym, włączonym pomiędzy C0 i S2 (napięcie osiąga chwilowe wartości równe Vcc, co powoduje świecenie).

Konfiguracja kontrolera LCD

Aby przygotować kontroler do pracy należy zdefiniować kilka podstawowych parametrów.

Należy dobrać podstawową częstotliwość pracy (ustalając częstotliwość f_{LCD} w module BasicTimer). W tym projekcie zastosujemy częstotliwość podstawową 256 Hz, co daje $f_{frame}=41$ Hz (częstotliwość podstawowa $256 \text{ Hz}/(2 * \text{MUX3})$), a więc nieco mniej od wymaganej. Zmniejszenie częstotliwości pozwoli dodatkowo zredukować pobór prądu.



Rys. 15. Mapa pamięci LCDMEM

Tab. 1. Zasady doboru przypisań wyprowadzeń

LCDP2	LCDP1	LCDP0	Wybrane wyprowadzenia
0	0	0	—
0	0	1	S0...S15
0	1	0	S0...S19
0	1	1	S0...S23
1	0	0	S0...S27
1	0	1	S0...S31
1	1	0	S0...S35
1	1	1	S0...S39

List. 3. Procedura przygotowująca sterownik LCD do pracy z wyświetlaczem

```

{
  BTCTL = BTDIV + BTIP0 + BTIP1; // ustawienie Basic Timera
  LCDCTL = LCDON + LCDSON + LCDMX1 + LCDP0 + LCDP1;
  // ustawienie kontrolera LCD
  P5SEL |= 0xfc; // ustawienie wyprowadzen
}
    
```

Należy zdefiniować wyprowadzenia, które będą sterować wyświetlaczem. Wyprowadzenia COM1...COM3 są współdzielone z wyprowadzeniami P5.2...P5.4, a wyprowadzenia R13...R33 (wejścia dla dzielnika rezystancyjnego) z P5.5...P5.7. Aby mogły one sterować LCD (realizować funkcję specjalną), należy w rejestrze P5SEL dokonać odpowiedniego wyboru. Z kolei mechanizm wyboru linii segmentów jest uproszczony.

Wyboru dokonuje się w rejestrze LCDCTL i wpływa on na zachowanie całej grupy linii. Należy pamiętać, że wszystkie wyprowadzenia z wybranej grupy tracą zdolność pracy jako zwykłe linie I/O.

Za włączanie sterownika odpowiada bit LCDCTL.LCDON. Włączenie kontrolera powoduje pojawienie się napięcia Vcc na wyprowadzeniu V1(R33) i zasilenie dzielnika rezystancyjnego. Ciekawostką jest możliwość wygaszenia LCD bez wyłączenia kontrolera przy użyciu bitu kontrolnego LCDSON. Wyświetlacz w prezentowanym przykładzie jest dołączony do wyprowadzeń COM0...COM2 i S0...S23, wybieramy więc czwarte ustawienie z wymienionych w tab. 1. Na list. 3

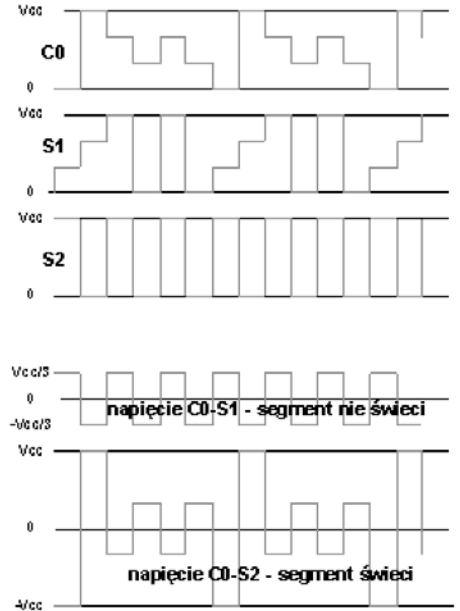
pokazano procedurę przygotowującą sterownik LCD do pracy z wyświetlaczem.

Podłączenie wyświetlacza do mikrokontrolera

Do uzyskania obrazu na wyświetlaczu wymaganych jest tylko kilka elementów zewnętrznych: rezonator kwarcowy oraz trzy rezystory dla generatora napięć. Ciekawostką jest, że sam mikrokontroler nie wymaga rezonatoru

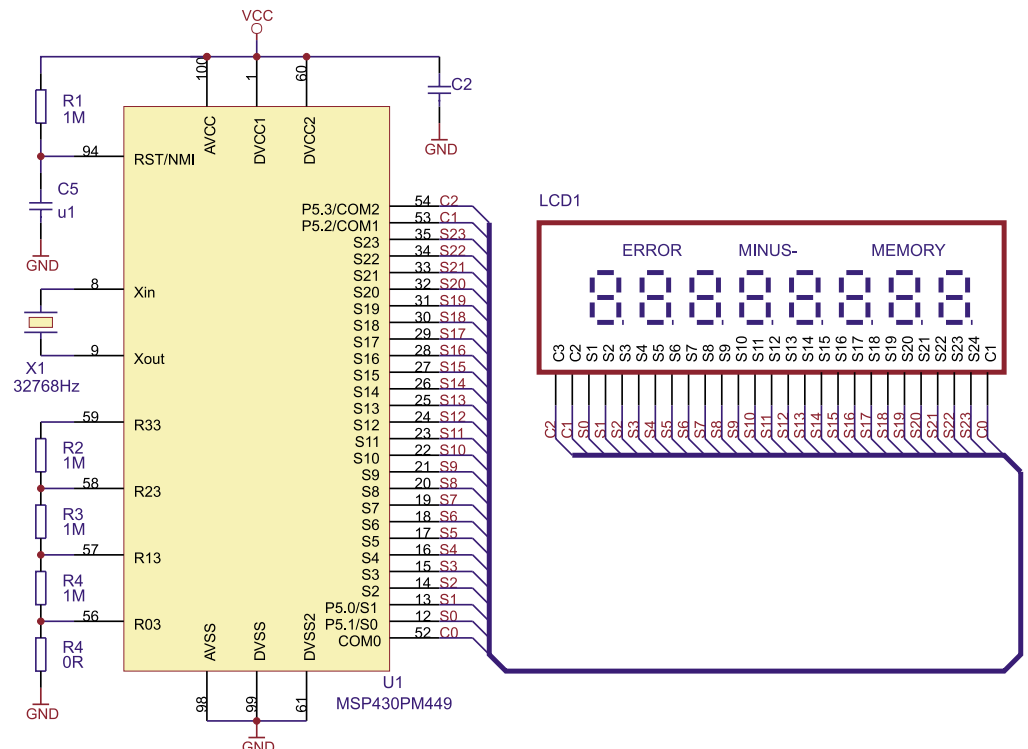
ra kwarcowego do pracy, jest on jednak niezbędny dla prawidłowego działania sterownika wyświetlacza.

Na rys. 17 zamieszczono schemat połączeń. Jest to fragment schematu płytki Demo449 (szczegółowo opisany w EP12/2008). Widoczne na schemacie złącze ma układ taki jak złącze LCD1 płytki Demo449, które dopasowano do wyświetlacza zastosowanego w zestawie. Wyświetlacz pochodzi z kalkulatora kupionego na targowisku (fot. 18). Oryginalnie wyświetlacz był podłączony do płytki drukowanej taśmą przewodzącą. Trzeba ją usunąć i zmyć resztki kleju spirytusem lub podobnym



Rys. 16. Przykładowe przebiegi sterujące

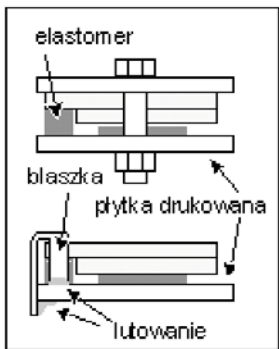
środkiem. Do podłączenia wyświetlacza potrzebny jest jeszcze elastomer. Wyświetlacz można przykryć odpowiednio przyciętą szybką z pleksi i przykręcić śrubami lub wykonać ramkę z cienkiej blaszki i przylutować ją do płytki (rys. 19). Należy pamiętać o zamontowaniu na płycce



Rys. 17. Schemat podłączenia wyświetlacza LCD do mikrokontrolera F449 w zestawie Demo449 (EP12/2008)



Fot. 18. Kalkulator zakupiony przez autora



Rys. 19. Sposoby mocowania wyświetlacza

Demo449 rezystorów R2, R3, R4 i R5. Zmieniając wartość rezystora R5 można wpływać na napięcie sterujące wyświetlacza, a w konsekwencji na kontrast.

Aby podłączyć inny typ wyświetlacza, można skorzystać ze złącza J9. Wymaga to oczywiście wykonania dodatkowej płytki drukowanej.

Wyświetlamy

Aby zrozumieć zasadę, wedle której na wyświetlaczu pojawiają się cyfry, prześledzimy drogę począwszy od budowy wyświetlacza, poprzez schemat połączeń, aż do programu. Każdemu z 67 elementów wyświetlacza przypisujemy oznaczenie składające się z numeru cyfry oraz litery określają-

Tab.2. Zasady identyfikacji elementu pola wyświetlacza i jego reprezentację w pamięci LCDM oraz buforze buffer_LCD

L.p.	Symbol	Segment.Podłoże	LCDMx.bit	Bufor[n].bit
1	1A	1.1	1.4	0.0
2	1B	2.1	2.1	0.1
3	1C	1.2	1.6	0.2
4	1D	0.2	1.2	0.3
5	1E	0.1	1.1	0.4
6	1F	0.0	1.0	0.5
7	1G	1.1	1.5	0.6
8	1DP	2.2	2.2	0.7
9	2A	4.0	3.0	1.0
10	2B	5.1	3.5	1.1
11	2C	5.2	3.2	1.2
12	2D	3.2	2.6	1.3
13	2E	3.1	2.5	1.4
14	2F	3.0	2.4	1.5
15	2G	4.1	3.1	1.6
16	2DP	5.2	3.6	1.7
17	3A	7.1	4.1	2.0
28	3B	8.1	5.6	2.1
...
63	8G	22.1	12.1	1.6
64	8DP	23.2	12.6	1.7
65	MINUS	17.0	9.4	8.0
66	MEMORY	20.0	11.0	8.1
67	ERROR	21.0	12.4	8.2

cej segment (za wyjątkiem znaków specjalnych), które znajdziemy w drugiej kolumnie tabeli tab. 2. Każdy z tych elementów włączony jest między jedno z doprowadzeń podłoża i segmentów (kolumna trzecia), co według schematu pamięci LCDMEM (rys. 15) pozwala określić jego lokalizację (kolumna czwarta). Ostatnim elementem jest przypisanie segmentowi miejsca w buforze. To z tego bufora pobiera dane funkcja obsługi wyświetlania.

Oprogramowanie

Wygodnym rozwiązaniem dla sterowania wyświetlaczem ciekłokrystalicznym jest założenie bufora dla danych graficznych. Bufor LCD_buffer ma rozmiar 9 bajtów. Każdy z ośmiu początkowych bajtów w buforze reprezentuje jed-

ną z cyfr wyświetlacza. W każdym bajcie znajdziemy wszystkie siedem segmentów cyfry oraz kropkę. Ostatni bajt zawiera bity znaków specjalnych (MEMORY, MINUS i ERROR). Funkcja niskiego poziomu seven_seg2LCD z parametrem w postaci wskaźnika na bufor graficzny dokonuje translacji zawartości bufora na obraz na polu wyświetlacza. Zastosowanie wskaźnika pozwala na użycie wielu buforów danych lub łańcuchów znaków zawartych w dowolnie zlokalizowanych tablicach. Program zawiera również definicje segmentów kodu siedmio-segmentowego oraz tablicę znaki o dziesięciu elementach stanowiących obrazy cyfr 0...9.

Algorytm działania funkcji seven_seg2LCD (list. 4) nie jest skompli-

List. 4. Funkcja seven_seg2LCD

```
void seven_seg2LCD (unsigned char* source)
{
    unsigned short data_src; //
    unsigned short data_eval;
    unsigned short round_cnt;
    unsigned char *pntr;
    pntr = (unsigned char*) LCDMEM;
    for (round_cnt = 0; round_cnt < 4; round_cnt++)
    { // cztery pary cyfr do analizy
        // każda para cyfr współpracuje z trzema komórkami LCDM
        data_eval=0;
        data_src = *source; //pobranie danych
        if (data_src & segA) data_eval |= LCDA1;
        if (data_src & segC) data_eval |= LCDC1;
        if (data_src & segD) data_eval |= LCDD1;
        if (data_src & segE) data_eval |= LCDE1;
        if (data_src & segF) data_eval |= LCDF1;
        if (data_src & segG) data_eval |= LCDG1;
        *pntr = data_eval; // pierwsza komórka wyjściowa
        data_eval=0;
        if (data_src & segB) data_eval |= LCDB1;
        if (data_src & segP) data_eval |= LCDP1;
        //zakonczenie analizy pierwszej cyfry w parze
        source++;
        data_src = *source; //pobranie danych
        if (data_src & segD) data_eval |= LCDDr;
        if (data_src & segE) data_eval |= LCDEr;
        if (data_src & segF) data_eval |= LCDFr;
        pntr++;
        *pntr = data_eval; // druga komórka wyjściowa
        data_eval=0;
        if (data_src & segA) data_eval |= LCDAr;
        if (data_src & segB) data_eval |= LCDBr;
        if (data_src & segC) data_eval |= LCDCr;
        if (data_src & segG) data_eval |= LCDGr;
        if (data_src & segP) data_eval |= LCDPr;
        pntr++;
        *pntr = data_eval; // trzecia komórka wyjściowa
        *source++; // analiza dwóch cyfr zakończona
        pntr++;
    } // osiem cyfr juz narysowane
    data_src = *source; //pobranie danych znakow specjalnych
    pntr--;
    if (data_src & ERROR) *pntr |= segERR;
    pntr--;
    if (data_src & MEMORY) *pntr |= segMEM;
    pntr-=2;
    if (data_src & MINUS) *pntr |= segMIN;
    //znaki specjalne ustawione
}
```

List. 5. Program wyświetlający znaki na LCD, korzystający z funkcji `seven_seg2LCD`

```
LCD_buffer[0] = znaki[1];
LCD_buffer[1] = znaki[2];
LCD_buffer[2] = znaki[3];
LCD_buffer[3] = znaki[4];
LCD_buffer[4] = znaki[5];
LCD_buffer[5] = znaki[6];
LCD_buffer[6] = znaki[7];
LCD_buffer[7] = znaki[8];
LCD_buffer[8] = MINUS;
seven_seg2LCD(LCD_buffer);
```

kowany. Dla uproszczenia zapisu wykorzystano fakt, że każda para z ośmiu cyfr wyświetlacza ma podobną budowę i jest reprezentowana w trzech sąsiednich komórkach

Na list. 6 pokazano przykład wyświetlania zmiennej łańcuchowej zlokalizowanej w pamięci kodu. W podobny sposób można wyświetlać dowolny łańcuch znaków liczbę, należy tylko pamiętać, przy przed wyświetleniem liczby wykonać jej konwersję na zmienną łańcuchową.

Załóżmy, że tablica `string[8]` zawiera elementy zbudowane tak, jak elementy tablicy `znaki`

List. 6. Przykład wyświetlenia zmiennej łańcuchowej zlokalizowanej w pamięci kodu

```
const unsigned char string[8] = {data1,data2,data3,data4,data5,data6,data7,data8,dataMINUS};
uni_ptr = (unsigned char *)string;
seven_seg2LCD(uni_ptr);
```

LCDM. Program wykonuje cztery pętle, w których analizuje zawartość dwóch komórek bufora wejściowego i wypełnia trzy komórki LCDM, rysując dwie cyfry. Na zakończenie uzupełnia jeszcze pamięć LCDM bitami znaków specjalnych.

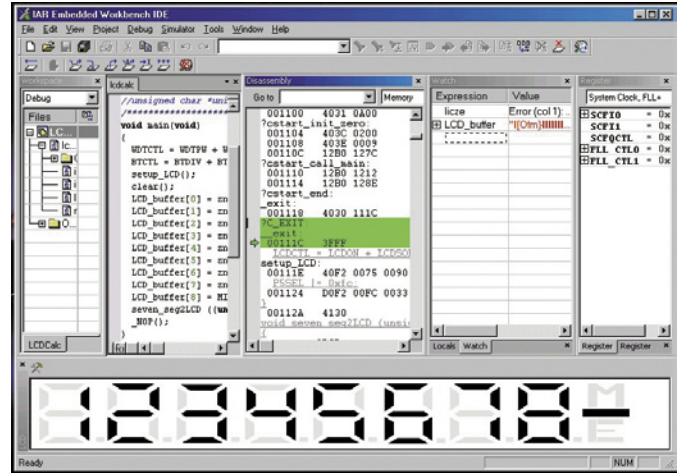
Najprostszy program zapisu do wyświetlacza korzystający z tak zbudowanej funkcji pokazano na list. 5.

W wyniku zadziałania powyższego fragmentu programu zostanie wyświetlony ciąg znaków: -12345678-.

– są to bitowe obrazy znaków przeznaczonych do wyświetlenia. Po wykonaniu programu, jak poniżej, ponownie zostanie wyświetlone wskazanie -12345678-.

Simulator wyświetlacza

W środowisku IAR Embedded Workbench, począwszy od wersji 3.0, wprowadzono możliwość symulacji zachowania wyświetlacza (rys. 20). Dostępne biblioteki obejmują niestety tylko dwa typy wyświetlaczy. W pakiecie brakuje również edytora



Rys. 20. Simulator LCD wbudowany w pakiet IAR Embedded Workbench

Workbench IDE w trybie debugowania lub symulacji i w menu `View` wybrać pozycję

`LCD Display`, a pojawi się pole podglądu wyświetlacza. Po kliknięciu na nie prawym klawiszem myszy należy wskazać właściwą bibliotekę. Podgląd treści wyświetlacza jest aktywny zarówno wtedy, gdy wybrana jest opcja pracy z symulatorem procesora, jak i w trybie pracy z „żywym” układem.

**Mariusz Kaczor
Contrans TI**

Dokumentacje projektów oraz inne materiały i informacje na temat mikrokontrolerów MSP430 są dostępne na stronie msp430.ep.com.pl.

R E K L

TONSIL sklep internetowy
zestawy hi-fi głośniki
www.e-tonsil.pl

A M A

LE nadajemy kształt elektronice
www.lcel.com.pl

- klawiatury
- obudowy
- materiały pomocnicze
- wsparcie technologiczne
- plasty czołowe
- akcesoria

www.alarmy-gerard.pl
SKLEP INTERNETOWY: www.gerard.pl

LARO www.laro.com.pl
CZĘŚCI ELEKTRONICZNE

sklep **INDUCTORS**.pl
ELEMENTY INDUKCYJNE
info@teyster.pl

GAMMA www.gamma.pl
PODZESPOŁY ELEKTRONICZNE
info@gamma.pl