



Dzisiejszy wykład, ostatni w podstawowym kursie BASCOM College, będzie miał trochę nietypowy, rzekłbym futurystyczny charakter. Zapoznamy się na nim z ostatnim „fajerwerkiem” opracowanym dla BASCOM-a przez MCS Electronics. Mój Przyjaciel Mark z podziwu godną konsekwencją realizuje swoją dewizę: „Co tu wymyślić, aby inni nie musieli myśleć”, rozbudowując coraz bardziej możliwości stworzonego przez siebie oprogramowania. Wszystko to sprawia, że kolejne zastosowania procesorów, będące dotąd terenem działania profesjonalistów, stają się dostępne nawet dla początkujących amatorów. To co niegdyś wymagało od zaawansowanego programisty wielu godzin ciężkiej pracy, obecnie możemy zrealizować za pomocą wydania jednego, prostego polecenia.

Nie ukrywam, że ostatnie opracowania Marka po raz kolejny „rzuciły mnie na kolana” i to nie tylko mnie. Mark tym razem „zatakował” jeden z najtrudniejszych problemów, z jakim spotykają się programiści: obsługę wyświetlaczy graficznych LCD, i zgodnie ze swoim zwyczajem sprowadził problem w zasadzie do jednego, nowego polecenia języka MCS BASIC. Jakby tego było mało, do BASCOM-a został dobudowany wyspecjalizowany moduł, którego zadaniem jest konwersja plików graficznych na format specjalnie opracowany dla potrzeb BASCOM-a: * .BGF, czyli BASCOM GRAPHICS FILE. Za chwilę przekonacie się, że problem ukazania na wyświetlaczu LCD dowolnego obrazu został sprowadzony do ... kilku kliknięć myszki i wydania jednego polecenia.

A czy wiecie, do czego to wszystko doprowadziło? Ano do tego, że jesteśmy w obecnej chwili w stanie zbudować komputer. Oczywiście nie będzie to maszyna, która mogłaby dorównać nawet PC XT, ale posiadająca wszystkie elementy „normalnego” komputera: procesor, klawiaturę i monitor z możliwością wyświetlania zarówno informacji w postaci tekstowej, jak i graficznej. Rzecz jasna, taki komputer nie nadawałby się do typowych

pracy wykonywanej przez te maszyny i nikomu nie przyszłoby nawet do głowy zastosować go np. do opracowywania tekstów. Jednak bez najmniejszych problemów możemy wykorzystać go w roli oscyloskopu o całym przystroju parametrach, analizatora stanów logicznych czy też warsztatowego „kombajnu” pomiarowego.

Chciałbym prosić Czytelników, którzy przeczytają ten wykład o zapoznanie się z treścią artykułów o oscyloskopie cyfrowym, opublikowanych w numerach 9, 10 i 11 Elektroniki Praktycznej lub o zajrzenie na stronę www.zmda.com. Mam nadzieję, że nasuną Wam się pewne skojarzenia i nowe pomysły ...

Podczas poprzednich wykładów zapoznaliśmy się z różnymi sposobami przekazywania informacji do i z procesora. Umieemy już posługiwać się prostą klawiaturą szesnastkową, bardzo dokładnie przedyskutowaliśmy problemy związane z obsługą alfanumerycznych wyświetlaczy LCD. Nie sądzę także, aby jakimkolwiek Studentowi BASCOM College mogła sprawić kłopot prezentacja wyników obliczeń dokonanych przez procesor na wyświetlaczach siedmiosegmentowych LED. Jednak wymienione sposoby przekazywania otoczeniu danych pochodzących z procesora mają jedną wspólną wadę: narzucają z góry typ znaków, jakie mają zostać ukazane na wyświetlaczu. Najmniejsze możliwości posiadają wyświetlacze LED, ponieważ korzystając z nich mamy tylko 10 cyfr do dyspozycji. Nieco większe jest pole zastosowań wyświetlaczy alfanumerycznych, które umożliwiają prezentację nie tylko cyfr, ale także wszystkich liter alfabetu łacińskiego i wielu znaków dodatkowych, w tym dowolnie definiowanych przez użytkownika. Jednak zawsze znaki wyświetlane będą w wyznaczonych wierszach i kolumnach, co praktycznie uniemożliwia „złożenie” z nich nawet najprostszego rysunku. Tymczasem, wielu konstruktorom marzy się możliwość pokazania na ekranie jakiegoś wyświetlacza informacji w formie, jaką możemy oglądać na monitorach komputerów: dowol-

nych obrazów graficznych, no i oczywiście także informacji tekstowej, ale przekazywanej za pomocą definiowanego zestawu znaków. Takie wyświetlacze zostały już dawno skonstruowane i noszą nazwę wyświetlaczy graficznych LCD. Z pewnością każdy z Was spotkał się już wielokrotnie z tymi elementami. Można je spotkać w masowo produkowanych „ręcznych” grach komputerowych, ale także w najnowocześniejszych komputerach klasy notebook. Marzeniem chyba każdego z Was jest posiadanie nowoczesnego, „płaskiego” monitora o fantastycznej jakości obrazu, w którym także zastosowany jest wyświetlacz graficzny LCD najnowszej generacji. Oczywiście, pomiędzy wyświetlaczem monochromatycznym 240*64 punktów a nowoczesnym monitorem komputerowym TFT istnieje przepaść technologiczna, ale obydwa te elementy działają na bardzo podobnej zasadzie.

Niestety, wyświetlacze graficzne LCD mają jeszcze jedną wspólną cechę: relatywnie wysoką cenę zakupu. Te właśnie elementy taniej stosunkowo powoli i to między innymi decyduje o wysokich cenach nateboków i monitorów LCD. Oczywiście, tak jak napisałem: ich cena jest **relatywnie** wysoka i zawsze należy ją odnieść do możliwości, jakie daje oceniany element. Dla przykładu: wyświetlacz 240*64 punkty, z jakim zapoznamy się za chwilę kosztuje, w zależności od dostawcy ok. 120PLN. Tylko w trybie tekstowym, pomijając jego możliwości graficzne umożliwia on wyświetlenie na ekranie 8 wierszy, każdy po 40 znaków, czyli razem 320 symboli alfanumerycznych. Popularny wyświetlacz LCD 16*1 znaków kosztuje ok. 18PLN, także w zależności od dostawcy. Policzcie teraz, który z tych elementów jest w rzeczywistości tańszy, bo mnie wyszło, że koszt wyświetlenia jednego znaku jest na wyświetlaczu graficznym trzykrotnie mniejszy, pomijając jego możliwości graficzne!

Zanim przejdziemy do omawiania konkretnych poleceń języka MCS BASIC,

za pomocą których będziemy mogli wyczerpać na ekranie graficznym dowolne kształty i figury, powiedzmy sobie parę słów o tym, czym właściwie są wyświetlacze graficzne LCD. Nie ma najmniejszego sensu wnikać w tej chwili w szczegóły ich budowy. Jesteśmy w tym momencie tylko programistami i dla nas wyświetlacz pracujący w trybie graficznym jest po prostu zbiorem punktów, z których każdy może być widoczny lub ukryty. Wyświetlacz, którym będziemy się posługiwać, jako jednym obsługiwany w obecnym momencie przez pakiet BASCOM, jest matrycą składającą się z 240 kolumn i 64 wierszy, czyli jego rozdzielczość wynosi 15360 punktów - pikseli, z których każdy może zostać indywidualnie zapalony lub wyłączony. Wyświetlacze graficzne, które możemy sterować z poziomu języka MCS BASIC muszą być zgodne z opracowanym przez firmę TOSHIBA standardem T6963C, którego nazwa pochodzi od układu scalonego - drivera sterującego wyświetlaczem.

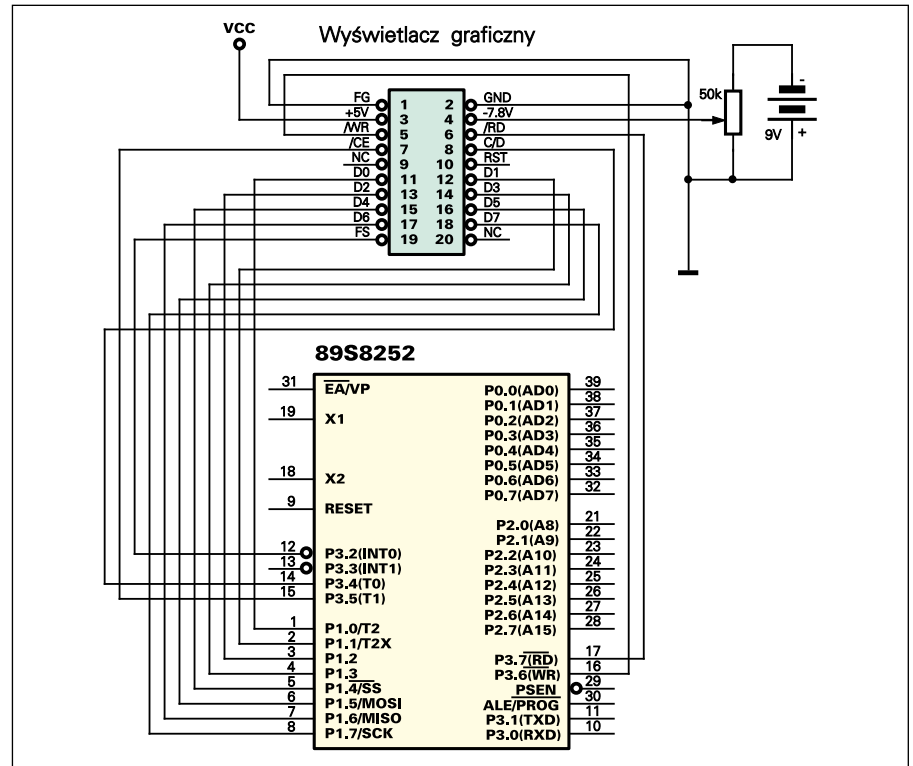
Wyświetlacze tekstowe, z którymi mieliśmy do tej pory do czynienia były dla procesorów stosunkowo „nóżkooszczędne”. Do ich sterowania wymagane było tylko sześć wyprowadzeń i dzięki temu podłączenie tych wyświetlaczy nawet do 20-wyprowadzeniowych procesorów nie narażało większych problemów. Zapomnijmy jednak o tych dobrych czasach w przypadku wyświetlaczy graficznych! Elementy te wymagają już podłączenia do aż 13 aktywnych wyprowadzeń procesora, co w zasadzie wyklucza ich stosowanie w praktycznych układach z procesorami 'X051. Nie oznacza to jednak, że nie możemy podłączyć wyświetlacza graficznego do naszej płytki testowej i dokonywać z nim dowolnie skomplikowanych eksperymentów. Możemy to uczynić nawet bez programowania procesora, ponieważ wyświetlacz taki może być sterowany także w emulacji sprzętowej. Jednak trudno sobie wyobrazić praktyczne zastosowanie układu, w którym będziemy dysponować zaledwie 2 wolnymi wyprowadzeniami procesora.

Podczas testowania nowych poleceń języka MCS BASIC korzystałem z procesora typu 89S8252, jednego z nielicznych chipów z rodziny '51, który może być programowany w systemie za pośrednictwem magistrali SPI, podobnie jak procesory AVR. Schemat podłączenia wyświetlacza typu TLX711A do tego procesora został pokazany na rysunku 1. Bez większych problemów możemy ten układ adaptować do wymagań układu z procesorami 'X051.

Wyświetlacze graficzne wymagają z zasady dodatkowego źródła napięcia zasilania, ujemnego względem masy układu. Niektóre z nich posiadają wbudowane przetwornice generujące takie napięcie, ale niestety nie dotyczy to popularnego wyświetlacza typu TLX711A. A zatem musimy do-

prowadzić do niego ujemne napięcie o wartości ok. 7,8V z osobnego źródła, którym podczas prób może być z powodzeniem zwykła bateria 9V, włączona zgodnie ze schematem z **rysunku 1**. Potencjometr montażowy pozwala na regulację kontrastu, podobnie jak w przypadku wyświetlaczy alfanumerycznych.

Rys. 1



Obsługa wyświetlacza graficznego LCD w języku MCS BASIC. Tryb tekstowy

Nie możemy zapominać, że prawie każdy wyświetlacz graficzny może pracować także jako wyświetlacz tekstowy, alfanumeryczny. Nasz przykładowy wyświetlacz typu TLX711A umożliwia prezentację aż ośmiu wierszy, z których każdy może składać się z 30 lub 40 znaków, w zależności od wybranego zestawu czcionek. Nasze doświadczenia z wyświetlaczem graficznym rozpoczniemy trochę przekornie: właśnie od jego pracy w trybie tekstowym. Polecenia obsługi wyświetlacza pracującego w tym trybie praktycznie nie różnią się od poleceń stosowanych do sterowania wyświetlaczem alfanumerycznym, z tym, że ich zestaw jest (jak na razie) nieco uboższy.

Do obsługi wyświetlacza graficznego pracującego w trybie tekstowym używane są następujące polecenia:

- CLS - czyszczenie całej zawartości wyświetlacza
- CLS TEXT - usuwanie z wyświetlacza tekstu, z pozostawieniem elementów graficznych

- CLS GRAPH - usuwanie z ekranu elementów graficznych z pozostawieniem tekstu
- LOCATE [wiersz, kolumna] - umieszczanie kursora na wskazanej pozycji
- CURSOR OFF, CURSOR ON usuwanie i umieszczanie kursora na ekranie

Napiszmy zatem pierwszy program testowy służący obsłudze wyświetlacza graficznego.

```

Scrystal = 8000000 'określenie częstotliwości
                                rezonatora kwarcowego
Config Graphlcd = 240 * 64 , Port = P1 , Ce = P3.5
, Cd = P3.4 , Cols = 30 'konfiguracja
                                sprzętowa wyświetlacza

Reset P3.2 ' znaki o szerokości 8
                                pikseli, 30 kolumn.

'Dodatkowy komentarz: wymuszenie stanu
niskiego na wejściu FS (Font Select - wybór czcionki)
wyświetlacza powoduje generowanie znaków o szerokości 8 pikseli. Stan wysoki na tym wejściu pozwala na wybranie mniejszych czcionek o szerokości 7 pikseli,
których w jednym wierszu zmieści się 40.

Cls 'wyczyść całą zawartość wyświetlacza
'Dalsze polecenia nie wymagają już dodatkowego
komentarza. W obecnej chwili brak jeszcze możliwości
definiowania polskich znaków diakrytycznych!

Locate 1 , 1
Lcd „ Witam Czytelników Elektroniki“
Locate 2 , 1
Lcd „dla Wszystkich na kolejnym wy.“
Locate 3 , 1
Lcd „kładzie BASCOM College. Lekcja „
Locate 4 , 1
Lcd „dzisiejsza będzie poświęcona „
Locate 5 , 1
Lcd „wyświetlaczom graficznym LCD“
Locate 6 , 1
Lcd „i ich sterowaniu z poziomu“
Locate 7 , 1
Lcd „języka MCS BASIC. Serdecznie“
Locate 8 , 1
Lcd „zapraszam na zajęcia!“
End
    
```

Efekt działania tego programu możemy zobaczyć na **rysunku 2**. Nie po to jednak zajęliśmy się wyświetlaczami graficznymi, aby prezentować na nich wyłącznie teksty. Przyjdźmy zatem do następnej części wykładu.

Rys. 2



Obsługa wyświetlacza graficznego LCD w języku MCS BASIC. Tryb graficzny

Obawiam się, że pewna część Czytelników z pewną obawą czekała na rozpoczęcie tej części wykładu. Obsługa wyświetlacza w trybie graficznym to coś zupełnie nowego, coś, co wymaga z pewnością poznania wielu nowych poleceń i skomplikowanych zasad ich stosowania. Tymczasem nic podobnego, w rzeczywistości do obsługi grafiki wykorzystywać będziemy tylko jedno polecenie:

Pset [oś X] , [oś Y] , [1,0]

Gdzie:

Oś X - wartość z zakresu 0 - 239
(dla wyświetlacza 240x64)

Oś Y - wartość z zakresu 0 - 63
(dla wyświetlacza 240x64)

1 - włącz określony współrzędnymi punkt
0 - wyłącz określony współrzędnymi punkt

I jest to już wszystko, co jest potrzebne do narysowania na wyświetlaczu całkowicie dowolnego rysunku. Cała sztuka tworzenia rysunków tą metodą polega na umiejętnym stosowaniu znanych już Wam poleceń języka MCS BASIC. Polecenie:

PSET 10, 10, 1

spowoduje włączenie jednego piksela na przecięciu wskazanych osi X i Y. Natomiast programik

```
FOR Z = 10 TO 20
PSET Z, 10, 1
NEXT Z
```

wykreśli nam na ekranie wyświetlacza krótką linię, o długości 10 pikseli, umieszczoną w odległości 10 punktów od osi X i zaczynającą się także w odległości 10 punktów od osi Y.

Jeżeli już nasyciliśmy się widokiem tak pięknie i równiutko narysowanej kreseczki, to zawsze możemy wydać następujący ciąg poleceń:

```
FOR Z = 10 TO 20
PSET Z, 10, 0
NEXT Z
```

Jaki będzie efekt wykonania drugiego programu? Zastanówcie się także, w jaki

sposób wykreślić na ekranie proste figury geometryczne: kwadrat, prostokąt, trójkąt czy wreszcie najtrudniejsze, czyli narysowanie koła, tak aby przypominało koło, a nie obgryzione jabłko. Jednak takie ćwiczenia będą miały wy łącznie poznawczy charakter, ponieważ w najbliższym czasie do MCS BASIC dodane zostaną kolejne polecenia, o działaniu podobnym do działania poleceń stosowanych w MS QBASIC. Będą to: CIRCLE, LINE, WINDOW i prawdopodobnie jeszcze kilka innych.

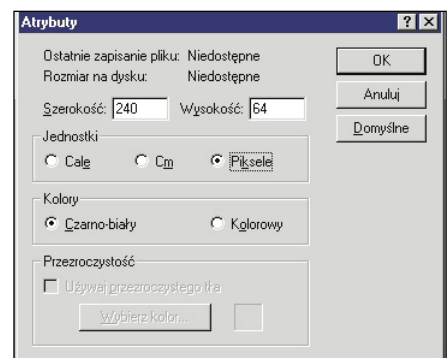
Obsługa wyświetlacza graficznego LCD w języku MCS BASIC. Prezentacja obrazów graficznych

Przejdźmy teraz do chyba najbardziej widowiskowej części dzisiejszego wykładu. Wiemy już, że za pomocą jednego tylko polecenia PSET możemy narysować na ekranie wyświetlacza LCD całkowicie dowolny rysunek. Po dodaniu nowych poleceń języka MCS BASIC, nie będziemy mieli najmniejszego problemu z rysowaniem figur geometrycznych bez konieczności pisania dość skomplikowanych podprogramów. Podejrzewam jednak, że Wasze apetyty będą rosły w miarę zjedzenia i zechcecie w przyszłości dokonać czegoś więcej. Możemy, na przykład, korzystając z opisanych i zapowiadanych poleceń, wykonać oscyloskop cyfrowy, analizator stanów logicznych czy też wyjątkowo efektywny analizator widma sygnału akustycznego. Jednak w ten sposób wykonane ekrany zawsze będą „matematyczne”, bez jakiegokolwiek ozdoby czy nawet efektywnego logo Waszej firmy. Budowanie grafiki „z prawdziwego zdarzenia” za pomocą polecenia PSET jest wprawdzie teoretycznie możliwe, ale potwornie pracochłonne i zajmujące wielkie obszary pamięci programu. Wychodząc takim potrzebom naprzeciw, firma MCS Electronics stworzyła narzędzie zupełnie wyjątkowe, za pomocą którego możemy błyskawicznie wysłać na ekran wyświetlacza dowolny obrazek. No może nie całkiem dowolny, musi to być obrazek monochromatyczny i zawierający nie więcej pikseli, niż mieści się na ekranie stosowanego wyświetlacza. Jednak jest to też coś, przeciwieństwo sztuki tworzenia grafiki polega także na sztuce upraszczania obrazu, tak aby zawierając jak najmniejszą liczbę elementów, przynosił zamierzone przez jego twórcę treści. Popatrzcie teraz na winięty tego artykułu, na którą być może nie zwróciliście jeszcze uwagi. Przedstawia ona zdjęcie wyświetlacza LCD, na którego ekranie został wyświetlony

obrazek. Nie jest to z pewnością dzieło Rembrandta, ale zawsze jest to jakaś grafika, która może być dobrym przykładem wykorzystania możliwości, jakie daje nam obecnie BASCOM. Jak jednak taki obrazek wykonać?

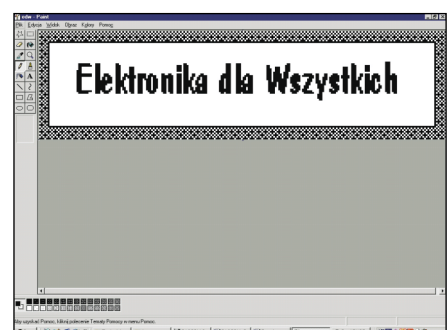
Metod jest wiele. Można, drogą kolejnych transformacji przerobić jakiś nieskomplikowany rysunek, tak aby „mieścił się” w zadanym obszarze 240x64 pikseli. Jednak jest to zadanie dość trudne i wymagające sporych zdolności plastycznych. Metodą prostszą jest wykonanie rysunku od początku, za pomocą edytora graficznego umożliwiającego edycję pojedynczych pikseli. A taki edytor znajduje się obecnie praktycznie na każdym komputerze pracującym pod kontrolą jednej z wersji MS WINDOWS. Mam tu na myśli zwykle pogardzanego (i słusznie) PAINTBRUSH-a, lub jego nowszą wersję: PAINT-a. Ten edytor, ze względu na znikome możliwości, wykorzystywany bardzo rzadko, jest właśnie idealnym narzędziem do przygotowania obrazka przeznaczonego do umieszczenia na wyświetlaczu graficznym LCD.

Nie mam zamiaru prowadzić tu kursu obsługi PAINT-a i podpowiem Wam tylko podstawowe zasady tworzenia takich obrazków. Po pierwsze, musimy na początku pracy określić wymiary rysunku w pikselach. Rozmiary te mogą być dowolne, byle nie większe od pola odczytowego posiadanego wyświetlacza, czyli w naszym przypadku 240x64 piksele (**rysunek 3**). Następnie, wykorzystując skromne możliwości oferowane nam przez PAINT-a, po prostu ... rysujemy to, co chcemy zobaczyć na ekranie. Może to być na przykład logo naszego pisma (**rysunek 4**).

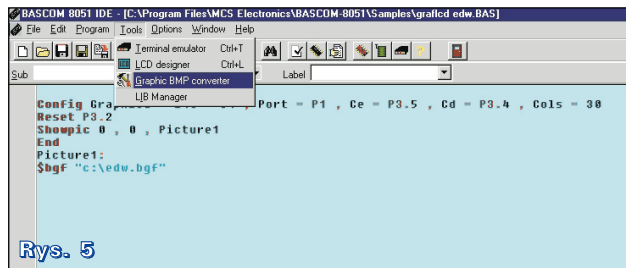


Rys. 3

Rys. 4



Gotowy plik zapisujemy w dowolnym katalogu i uruchamiamy naszego BASCOM-a. Otwieramy panel TOOLS i wybieramy z niego nowe narzędzie: GRAPHIC BMP CONVERTER (rysunek 5), za pomocą którego możemy dokonać konwersji utworzonego przez PAINT-a pliku BMP na nowy format opracowany przez MCS Electronics - BGF. Odczytujemy za pomocą funkcji LOAD zapisany uprzednio plik BMP i po ułamku



Rys. 5



Rys. 6

sekundy mamy już przekonwertowany rysunek, gotowy do umieszczenia na wyświetlaczu (rysunek 6). Musimy jedynie zapisać go w jakimś katalogu i napisać prosty program, który spowoduje pojawienie się naszego obrazka na wyświetlaczu. Teraz dopiero będziemy mogli ocenić, jakie rezultaty przyniosła praca Marka nad nowymi modułami pakietu BASOM. Aby wysłać dowolny, zapisany na dysku

obrazek, wykonany w formacie BGM, wystarczy wydać tylko jedno polecenie:

SHOWPIC [x], [Y], [nazwa pliku]

Gdzie:

X -, Y współrzędne pod jakimś musi znaleźć się górny lewy róg obrazka

Nazwa pliku - nazwa pliku BGM, w którym zapisany został obrazek, wraz ze ścieżką dostępu do niego

Cały kompletny program, którego wykonanie spowoduje rezultat pokazany na rysunku 7, wygląda następująco:

```
Config Graphlcd = 240 * 64, Port = P1, Ce = P3.5, Cd = P3.4, Cols = 30
Reset P3.2
Cls
Cursor Off
Showpic 0, 0, Picture1
End
Picture1:
$bgf 'c:\edw.bgf'
```

i nie sądzę, aby wymagał on jakichkolwiek komentarzy.



Rys. 7

To, co zrobiliśmy do tej pory było jedynie efektywnym pokazem nowych możliwości BASCOM-a, jednak bez zastosowania praktycznego. Takie zastosowanie znajdzie się z pewnością w przyszłości, np. podczas konstruowania przyrządów pomiarowych. Obrazki umieszczane w odpowiednich miejscach ekranu wyświetlacza mogą ułatwić korzystanie z urządzenia i uatrakcyjnić jego obsługę.

Zbigniew Raabe
e-mail: zbigniew.raabe@edw.com.pl

Podsumowanie kursu **BASCOM College**

Zgodnie z zapowiedzią sprzed miesiąca, niniejszy odcinek BASCOM College jest ostatnim w ramach regularnego kursu. Potem będziemy zapraszać na zajęcia fakultatywne. W kolejnych numerach EdW będziemy zamieszczać dalsze artykuły dotyczące BASCOM-a i procesorów, które można programować z pomocą różnych wersji programu. Nie będzie to jednak systematyczny cykl, tylko materiały obejmujące rozmaite zagadnienia i nowości związane z BASCOM-em. Zapraszamy też Czytelników do nadsyłania swoich projektów oraz wszystkiego, co wiąże się BASCOM-em i programowaniem z jego pomocą.

Dokładnie przed rokiem rozpoczęliśmy zajęcia i okazało się, że zainteresowanie kursem jest ogromne. Na początku każdy tydzień przynosił kilkaset zgłoszeń do BASCOM College. W sumie napłynęło 6000 zgłoszeń. Wielu Czytelników, także tych, którzy wcześniej nie mieli żadnego kontaktu z mikroprocesorami, z radością rozpoczęło naukę programowania. Po kilku miesiącach pojawiły się pierwsze sukcesy w postaci samodzielnych programów. Kolejne odcinki cyklu pozwoliły „szlifować formę”, rozwiązywać napotkane problemy oraz tworzyć coraz ciekawsze projekty. Ogłoszona niedawno ankieta pokazała, że tylko nieliczni „zgubili się” i pozostali w tyle. Zdecydowana większość uczestników realizowała program kursu na bieżąco, osiągając większe lub mniejsze sukcesy. Jednym z efektów jest artykuł z serii *µProjekty-3000*, zamieszczony na stronie 24. Autor napisał: *Jestem studentem „BASCOM College”.* *Pod wpływem artykułów ukazujących się ostatnio w EdW zainteresowałem się techniką mikroprocesorową. Niniejszy list jest tego efektem. Prezentowane urządzenie jest elektroniczną maszyną do pisania, której do pracy potrzebna jest drukarka i klawiatura PC.*

Studentów BASCOM College, mogących poszczycić się podobnymi projektami jest wielu. Zgodnie z zapowiedzią, podaną na początku kursu, chcielibyśmy oficjalnie potwierdzić zdobyte umiejętności i rozdać dyplomy.

Praca dyplomowa

Wszyscy studenci BASCOM College mogą ubiegać się o dyplom. Aby otrzymać dyplom ukończenia kursu BASCOM College należy wykonać pracę dyplomową. Podobnie jak w szkole i na uczelni, temat można wybrać samodzielnie. Tematem pracy musi być jakiegokolwiek urządzenie zawierające mikroprocesor 89CX051, a program ma być stworzony z pomocą pakietu BASCOM. Praca powinna zawierać następujące elementy:

- schemat ideowy układu
- krótki opis funkcji spełnianych przez układ
- program bądź kluczowe fragmenty programu.

Temat nie musi być trudny, nie jest też wymagane wykonanie modelu. Pomysł można zaczerpnąć z lekcji i ćwiczeń kursu. Nie trzeba się silić na wymyślanie nowych układów, raczej zachęcamy do tzw. reengineeringu, czyli przeprojektowania znanych z literatury układów na ich postać mikroprocesorową.

Kto chciałby wykonać działający model, może to zrobić. Nie jest to wymagane do otrzymania dyplomu, ale najlepsze modele po zaopiniowaniu i sprawdzeniu będą opublikowane na łamach EdW, a ich twórcy otrzymają pełne honoraria autorskie. Prace, obowiązkowo oznaczone dopiskiem **DYPLOM**, należy nadsyłać do 10 marca 2001 roku.