

# Sterownik wyświetlacza graficznego z RS232

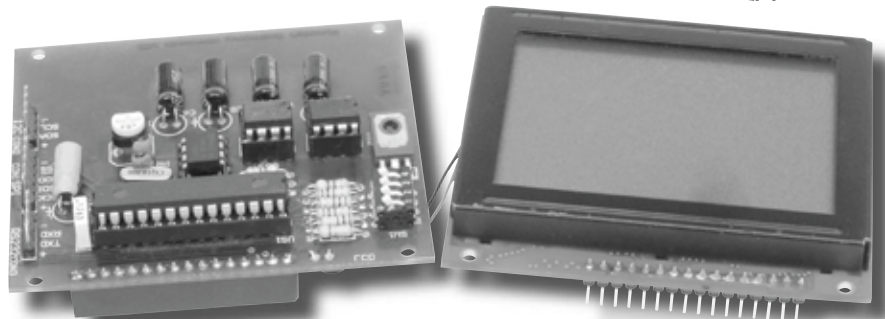
## AVT-544

Wyświetlacze graficzne znajdują coraz częściej zastosowanie w urządzeniach, w których dotychczas wystarczające były wyświetlacze alfanumeryczne, czy nawet wyświetlacze cyfrowe. Przyczyna takiej sytuacji wynika z coraz większych wymagań użytkowników tych urządzeń.

Najlepszym przykładem są przemiany dotyczące telefonów komórkowych, gdzie w przeciągu kilku lat wyświetlacze alfanumeryczne zostały zastąpione kolorowymi wyświetlaczami graficznymi. Zastosowanie wyświetlacza w układach elektronicznych wymaga użycia dużej ilości wyprowadzeń mikrokontrolera sterującego, a dodatkowo, w zależności od typu wyświetlacza, dużej szybkości pracy.

### Rekomendacje:

Proponowane rozwiązanie zainteresuje wszystkich użytkowników wyświetlaczy LCD. Stosunkowo łatwo jest znaleźć dla niego praktyczne zastosowanie. Można się więc spodziewać, że wielu czytelników uzna przedstawioną propozycję za atrakcyjną i wartą wykorzystania.



Wyświetlacze graficzne można podzielić na dwie grupy: z wbudowanym generatorem znaków lub bez generatora. Wyświetlacze z wbudowanym generatorem znaków zapewniają łatwiejsze sterowanie, jednak są znacznie droższe. W wyświetlaczu takim zastosowany procesor wyświetla znaki bezpośrednio po podaniu bajta danych odpowiadającej wartości wybranego znaku ASCII. Wyświetlacz taki może pracować także w trybie graficznym. Inaczej sprawa wygląda w uproszczonych wyświetlaczach bez generatora znaków, gdyż do wyświetlania jakiegokolwiek znaku niezbędne jest „zapalenie” odpowiednich pikseli odpowiadających za wyświetlenieżądanego znaku. W takiej sytuacji oprogramowanie mikrokontrolera sterującego takim wyświetlaczem musi być wyposażone w generator znaków. A wyświetlenie jednego znaku nie ogranicza się do wysłania jednego bajta określającego wartość liczbową ASCII lecz konieczne jest wysłanie co najmniej pięciu bajtów (w praktyce jeszcze więcej), jeżeli znak jest definiowany na matrycy 5x7, dla znaków o większych rozmiarach liczba wysłanych bajtów jest odpowiednio większa. Jak widać zapełnienie całego wyświetlacza tekstem wymaga stosunkowo dużo czasu, a w połączeniu z dużą liczbą wy-

maganych wyprowadzeń mikrokontrolera (typowo 13) sterowanie wyświetlaczem znacznie obciąża procesor, który ma pełnić jeszcze inne funkcje. Rozwiązaniem może być zastosowanie wyświetlacza posiadającego własny sterownik, który w celu ograniczenia liczby potrzebnych wyprowadzeń powinien być wyposażony w interfejs szeregowy. Pozwoli to „odciążyć” procesor zawarty w budowanym urządzeniu, a dodatkowo wykorzystać wyprowadzenia procesora do innych funkcji. Przykładem takiego sterownika jest układ prezentowany w artykule. Sterownik przystosowany jest do współpracy z wyświetlaczem graficznym typu JM12864A. Wyświetlacz ten posiada organizację 128x64 pikseli bez generatora znaków i jest wyposażony w sterowniki matrycy LCD typu: KS0107B i KS0108B. Dodatkowo wyświetlacz ten ma matryce diod świecących podświetlających pole odczytowe.

Komunikacja ze sterownikiem odbywa się poprzez interfejs szeregowy. Przy czym, dla zwiększenia uniwersalności możliwa jest komunikacja w trzech popularnych standardach:

- szeregowy asynchroniczny, zgodny z RS232, z możliwością ustawienia prędkości transmisji w zakresie 9600...115200 bd. W tym trybie wyświetlacz można dołączyć

**Tab. 1. Podstawowe parametry sterownika**

- Organizacja 128x64 piksele
- Matryca LCD podświetlana diodami LED
- Możliwość zdalnej regulacji intensywności podświetlenia
- Wbudowany generator znaków ASCII
- Rozmiar czcionki 5x7 lub 10x14
- Wbudowana funkcja rysowania linii
- Wbudowana funkcja rysowania pola
- Wbudowana funkcja rysowania koła
- Komunikacja poprzez interfejs: RS232, I2C lub SPI
- Pamięć EEPROM umożliwiającą zapisanie 32 gotowych obrazów
- Procesor sterujący taktowany zegarem 40 MHz
- Napięcie zasilania +5 V (150 mA z włączonym podświetleniem)

bezpośrednio do mikrokontrolera sterującego lub poprzez prosty tranzystorowy konwerter napięć do złącza szeregowego komputera.

- I2C z możliwością wybrania jednego z czterech adresów układu magistrali, co pozwala na dołączyć kilku układów do tej samej magistrali

- SPI z wykorzystaniem linii wyboru układu (*Slave Select*) – wykorzystywane są trzy linie, lub bez wyboru układu – komunikacja

Sterownik umożliwia także zdalne sterowanie podświetleniem wyświetlacza; oprócz jego włączenia lub wyłączenia możliwe jest także ustawienie intensywności podświetlenia. Intensywność może być regulowana w dziesięciu krokach i odbywa się poprzez modulację PWM z wykorzystaniem sterownika sprzętowego.

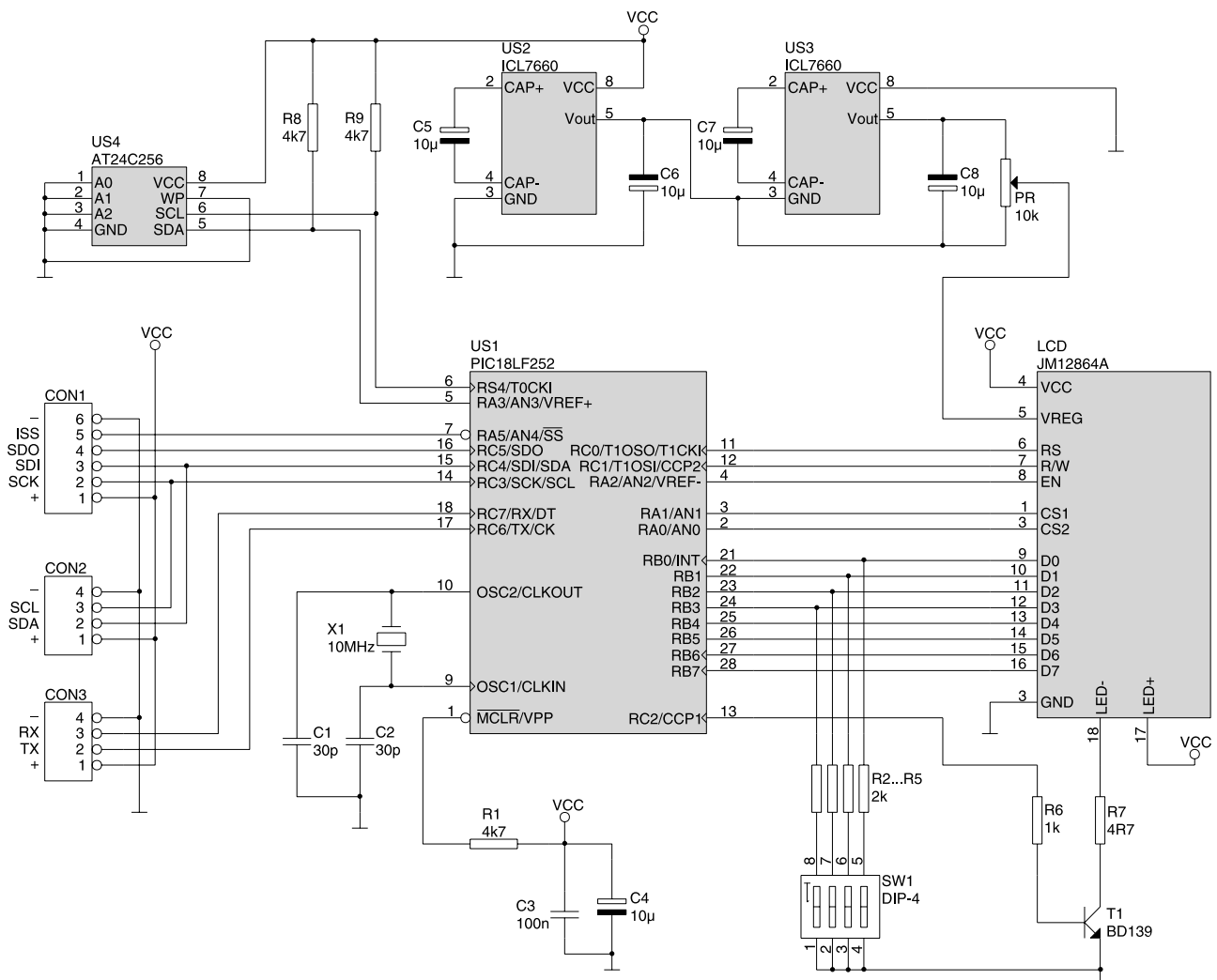
Dodatkowo w układzie sterownika wyświetlacza zastosowana została pamięć EEPROM o pojemności 32kB, służąca do zapisania

odbywa się po dwóch liniach (zegarowej i danych)

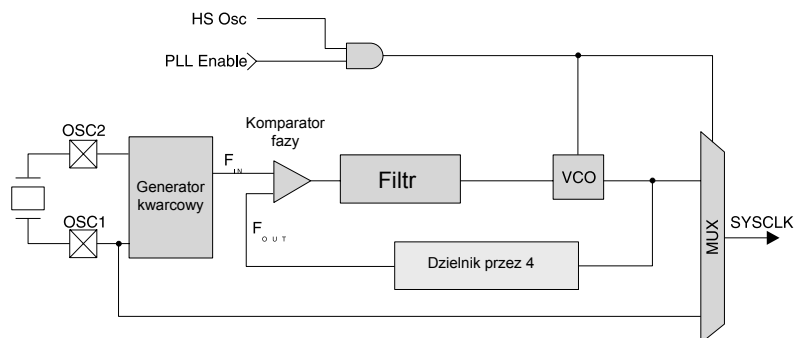
W każdym rodzaju transmisja odbywa się jednokierunkowo (z układu sterującego do wyświetlacza) - zalecana maksymalna częstotliwość sygnału zegarowego wynosi 100kHz (dla RS232 - 115200).

gotowych obrazów. Można w niej zapisać maksymalnie 32 przygotowane wcześniej informacje tekstowe lub graficzne, które następnie mogą zostać wyświetlone. Pamięć ta może być szczególnie przydatna, gdy wyświetlane będą elementy graficzne. Jeden obraz wyświetlacza składa się z 1kB pikseli, a wyświetlanie obrazu wymaga podania przez moduł współpracujący z wyświetlaczem informacji o stanie każdego piksela. Ustawienie stanu takiej liczby pikseli zajmuje dużo czasu, a dodatkowo jeśli wyświetlacz ma współpracować z procesorem zajmuje znaczną część jego pamięci programu. Natomiast do pamięci, w którą wyposażony jest sterownik można zapisać gotowe obrazy, które później mogą zostać odtworzone poprzez wydanie krótkiej komendy.

Sterownik umożliwia generowanie znaków w dwóch rozmiarach 5x7 i 10x14, dodatkowo ma funkcje rysowania linii, pola (np.



Rys. 1. Schemat elektryczny sterownika



Rys. 2. Budowa modułu PLL umożliwiającego zwielokrotnienie częstotliwości rezonatora kwarcowego

kwadratu), który może być wypełniony lub nie, a także koła. Koło również może zostać wypełnione lub pozostać puste, w wyniku czego powstanie okrąg. Z uwagi na stosunkowo małą rozdzielczość wyświetlacza otrzymane koło nie ma idealnych kształtów. Efekt ten pogłębia fakt, że pojedynczy piksel nie jest idealnym kwadratem, a prostokątem o rozmiarach 0,35 mm x 0,55 mm.

Wszystkie polecenia sterujące wyświetlaczem zorganizowane są w postaci komend z odpowiednimi parametrami, których znaczenie zostanie opisane w dalszej części artykułu.

Podstawowe parametry sterownika znajdują się w **tab. 1**.

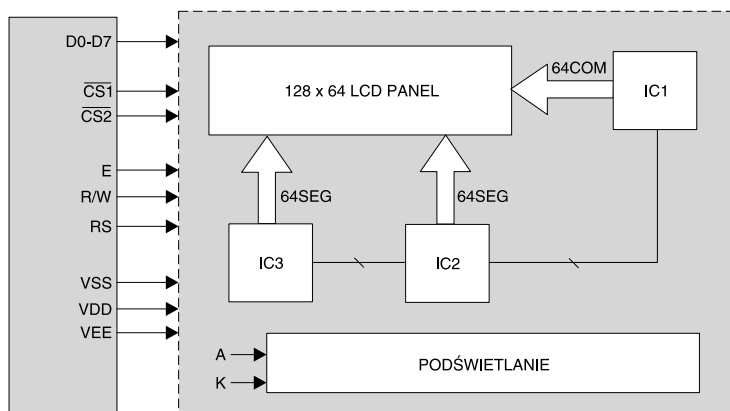
## Budowa

Schemat elektryczny sterownika jest przedstawiony na **rys. 1**. Jako układ sterujący został zastosowany mikrokontroler typ PIC18F252, który ma 32 kB pamięci programu, 1536 B pamięci RAM oraz 256 kB pamięci EEPROM. Sygnał zegarowy wytwarzany jest za pomocą zewnętrznego rezonatora kwarcowego, jednak pomimo zastosowania kwarcu o wartości 10 MHz, procesor tak naprawdę pracuje z częstotliwością 40 MHz. Spowodowane to jest tym, że wewnątrz układu znajduje się pętla PLL umożliwiająca pomnożenie częstotliwości oscylatora przez cztery, co właśnie zwiększa częstotliwość taktowania procesora do 40 MHz (na **rys. 2**). W ten sposób czas wykonania jednej instrukcji wynosi 100 ns, co w połączeniu z architekturą RISC, na której opiera się budowa procesora daje dużą moc obliczeniową. Duża prędkość pracy mikrokontrolera wpływa korzystnie na czytelność wyświetlanych komunikatów i grafiki, gdy jest ona

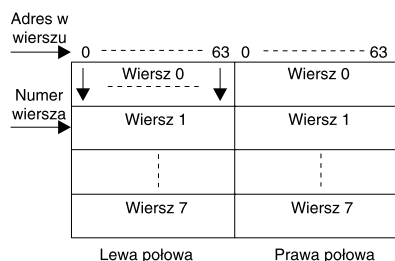
często aktualizowana. Zerowanie procesora przy włączeniu zasilania realizuje wewnętrzny układ resetu, dlatego wejście zerujące !MCLR jest na stałe podłączone do plusa zasilania poprzez rezystor R1. Wyświetlacz został podłączony do procesora poprzez port RB (linie danych), natomiast dodatkowe linie sterujące przez porty RA i RC. Z uwagi na brak wystarczającej liczby portów przełącznik SW1 został podłączony równolegle z wyświetlaczem do linii portu RB<3:0> i poprzez rezystory R2...R5 umożliwia wymuszenie stanu niskiego. Jednocześnie rezystory te nie „przeszkadzają” w komunikacji z wyświetlaczem. Odczyt stanu przełącznika odbywa się zawsze po włączeniu zasilania, przez włączenie wewnętrznych rezystorów podciągających do plusa zasilania, a następnie odczyt stanu portu RB. Jeśli odczytany stan będzie równy jeden, to dany styk przełącznika jest rozarty. Zwarcie styku przełącznika poprzez odpowiedni rezystor (R2...R5) wymusi stan niski. Ze względu na tę rezystancję stan ten nie odpowiada dokładnie wartości 0 V, lecz wynika z powstałego dzielnika ( $R_{wewnętrzne}/R2$ ).

Z uwagi jednak na fakt, że wewnętrzna rezystancja rezystorów podciągających ma wartość rzędu kilkudziesięciu kΩ, uzyskane napięcie mieści się w zakresie zera logicznego. Do sterowania podświetlaniem wyświetlacza zastosowano wzmacniacz tranzystorowy (T1), który z kolei jest sterowany poprzez wyjście portu RC2. Port ten oprócz typowych funkcji jest wyjściem sprzętowego sterownika PWM, dzięki czemu intensywność podświetlania może być regulowana na drodze cyfrowej szerokością impulsu sterującego, co z kolei umożliwia jej zdalną regulację.

Do poprawnej pracy wyświetlacza konieczne jest dostarczenie do wejścia układu regulacji kontrastu napięcia o wartości równej około -9 V. Napięcie to zostało uzyskane z popularnej przetwornicy napięcia ujemnego typu ICM7660. Z uwagi na fakt, że układ ten generuje napięcie ujemne o wartości zbliżonej do jego napięcia zasilania, uzyskane napięcie ma wartość -5 V. Aby uzyskać napięcie -9 V zastosowane zostały dwa takie układy połączone kaskadowo. Pierwszy z nich (US2) jest zasilany napięciem 5 V i na jego wyjściu otrzymuje się napięcie równe -5 V. Drugi układ (US3) jest zasilany również napięciem o wartości 5 V jednak z przesuniętym punktem masy. Dla układu US3 plus zasilania podłączony jest do masy całego sterownika, natomiast minus zasilania do napięcia -5 V względem masy otrzymanej z układu US2, w efekcie czego na jego wyjściu uzyskuje się napięcie o wartości około -10 V. Napięcie to jest podawane do wejścia układu regulacji kontrastu wyświetlacza poprzez potencjometr PR umożliwiającą jego odpowiednie ustawienie.



Rys. 3. Budowa wewnętrzna wyświetlacza typu JM12864



Rys. 4. Organizacja pamięci wyświetlacza LCD

Dołączona do procesora pamięć EEPROM służy do zapisania, a następnie odtworzenia całego pola wyświetlacza. Zastosowana pamięć typu AT24C256 (32 kB) umożliwia zapisanie 32 obrazów wyświetlacza. Na złączach CON1...CON3 zostały wprowadzone wszystkie sygnały potrzebne do komunikacji zgodnej z interfejsami: RS232, I2C, SPI, a także linie zasilania sterownika i wyświetlacza, jednak z uwagi na komunikację jednostronną nie wszystkie są wykorzystywane.

Na złączu CON1 znajduje się sygnał wyjścia danych RS232 „TX” jednak nie jest on wykorzystywany ze względu na transmisję jednokierunkową (tylko w stronę wyświetlacza). Również złącze transmisji SPI zawiera sygnał danych wyjściowych „SDO”, który nie jest wykorzystywany. Dodatkowo, jeśli przy transmisji SPI nie będzie używane wejście !SS, to do złącza CON2 przystosowanego do transmisji I2C może zostać dołączony układ nadrzędny, gdyż wyprowadzenia sprzętowego sterownika SPI i I2C są połączone do tych samych wyprowadzeń procesora. W tym przypadku linia SCL będzie linią zegarową, a linia SDA linią danych transmisji SPI.

### Sterowanie wyświetlaczem

Zastosowany wyświetlacz graficzny tak naprawdę składa się z dwóch połączonych bloków, z których jeden reprezentuje lewą, a drugi prawą połowę wyświetlacza. Budowa wewnętrzna wyświetlacza jest przedstawiona na rys. 3. Układ IC1 steruje kolumnami równolegle połączonych połówek, układ IC3 segmentami lewej połowy wyświetlacza, natomiast IC2 prawą połowę. Każda połowa zorganizowana jest w osiem wierszy, każdy wiersz ma szerokość ośmiu pikseli i długość 64 pik-

seli (rys. 4). Rozpoczynając zapis od początku pamięci wyświetlacza (wiersz 0) bajty zapisywane są kolejno od lewej do prawej strony wybranej połowy wyświetlacza. Po osiągnięciu maksymalnej wartości dla danego wiersza (63) kolejne dane będą zapisywane od początku następnego wiersza.

Taka organizacja nieco utrudnia wyświetlanie informacji w dowolnym miejscu ekranu. Aby włączyć jeden piksel należy podać jego położenie na osi X(0...127) i Y(0...63). Na tej podstawie procesor sterujący musi określić, w której połowie znajduje się dany piksel, następnie trzeba określić numer wiersza i na końcu numer wybranego punktu (piksela). Jak widać odszukanie właściwego miejsca na wyświetlaczu zajmuje sporo czasu, jednak to jeszcze nie koniec trudności, ponieważ wszystkie operacje zapisu i odczytu danych wykonywane są na całych bajtach, a nie na poszczególnych bitach. Dlatego aby zmienić wartość jednego bitu -odpowiadającego danemu pikselowi należy odczytać z wyświetlacza cały bajt z obliczonej wcześniej pozycji, zmodyfikować odpowiedni bit (np. funkcją AND lub OR) i tak zmodyfikowany bajt zapisać do wyświetlacza. Wykonanie takiej operacji dla całej pamięci wyświetlacza (8196 bitów) zajmuje na tyle dużo czasu, że obraz w prawym dolnym rogu pojawi się z zauważalnym opóźnieniem. Aby zapobiec efektowi opóźnienia w sterowniku zastosowana została tablica w pamięci RAM procesora będąca dokładną kopią danych wysyłanych do wyświetlenia. W ten sposób powstały dwa obszary pamięci obrazu: jedna pamięć wyświetlacza, której zawartość jest bezpośrednio widoczna w postaci zapalonych pikseli na matrycy LCD oraz pośrednia pamięć-procesora, której modyfikacja nie jest widoczna bezpośrednio na wyświetlaczu. Takie rozwiązanie umożliwia podczas wyświetlania jednego obrazu (pobieranego z pamięci wy-

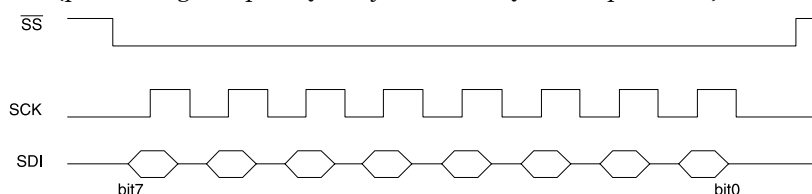
Tab. 2. Stan przełącznika SW1 i odpowiadające mu tryby pracy wyświetlacza

RS232 S1=0, S2=0		
S3	S4	Prędkość
0	1	9600 b
1	0	19200 b
0	1	57600 b
1	1	115200 b
I2C S1=1, S2=0		
S3	S4	Adres
0	0	90h
1	0	92h
0	1	94h
1	1	96h
SPI S1=0, S2=1 -!SS		
Komunikacja SPI z użyciem wejścia Slave Select.		
Stan przełącznika S3 i S4 nieistotny.		
SPI S1=1, S2=1		
Komunikacja SPI bez użycia wejścia Slave Select.		
Stan przełącznika S3 i S4 nieistotny.		

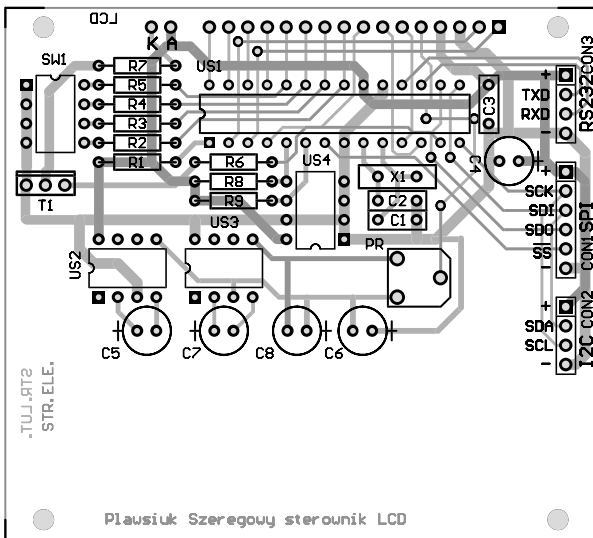
świetlacza) całkowitą modyfikację pamięci zawartej w procesorze, a następnie w dowolnym momencie przekopiowanie całej zawartości pamięci procesora do pamięci wyświetlacza. Dzięki temu „w ukryciu” można stworzyć obraz zawierający różne składniki (tekst, grafika), a następnie za pomocą jednej komendy przeniesić całą zawartość na wyświetlacz. Proces kopiowania zawartości pamięci procesora do pamięci wyświetlacza wykonywany jest na całych bajtach, bez konieczności wcześniejszego odczytu jakichkolwiek danych z wyświetlacza, dzięki czemu odświeżenie zawartości całego wyświetlacza polega na skopiowaniu 1kB danych, co następuje na tyle szybko, że nie występuje efekt opóźnienia pomiędzy danymi wyświetlonymi na początku (lewy górny róg) i na końcu wyświetlacza (prawy dolny róg).

### Realizacja transmisji szeregowej

Aby spełnić wszystkie wymagania czasowe w przestawionym układzie został zastosowany mikrokontroler, który wszystkie operacje związane z transmisją szeregową wykonuje sprzętowo. Takie rozwiązanie powoduje znaczne



Rys. 5. Przebiegi wymagane do poprawnej pracy wyświetlacza w trybie SPI



Rys. 6. Rozmieszczenie elementów na płytce sterownika

przyspieszenie jego pracy. Wynika to z faktu, że mikrokontroler nie musi odbierać poszczególnych bitów transmitowanych danych, ale jest informowany o odebraniu całego bajta przez sterownik sprzętowy. Dzięki temu ma więcej czasu na analizę danych i obsługę wyświetlacza. Dane są przetwarzane jednakowo dla wszystkich rodzajów interfejsów, a różny jest tylko sposób „składania” bitów w jeden bajt, jednak od strony programowej nie ma to znaczenia, ponieważ tym zajmują się moduły sprzętowe. Po odebraniu całego bajta generowane jest przerwanie, w którym następuje przepisanie odebranego bajta do bufora.

Do odbioru danych w trybie RS232 zastosowany został sprzętowy sterownik transmisji szeregowej RS232. Transmisja odbywa się w sposób asynchroniczny bez bitu parzystości z programowalną prędkością: 9600 bd, 19200 bd, 57600 bd lub 115200 bd.

Odbiór danych w standardzie SPI odbywa się przez moduł MSSP (ang. Master Synchronous Serial Port). Ponieważ dane są wysyłane tylko do wyświetlacza, więc wykorzystywane jest tylko wejście danych szeregowych SDI. W takt sygnału zegarowego podawanego przez układ nadrzędny poszczególne bity zapisywane są do rejestru SSPSR, a po odebraniu całego bajta następuje jego przepisanie do rejestru SSPBUF i wygenerowany zostaje sygnał przerwania, aby jednostka cen-

tralna mikrokontrolera mogła przetworzyć odebrany bajt. W opisywanym układzie sterownik SPI został skonfigurowany do pracy z narastającym zboczem sygnału zegarowego, to znaczy stan wejścia SDI jest zapisywany do rejestru SSPSR w momencie zmiany stanu z niskiego na wysokie na wejściu SCK. Dodatkowo istnieje możliwość wyboru, czy dane mają być odbierane przez cały czas, czy dopiero po wybraniu układu (stan niski na wejściu !SS), co umożliwia podłączenie kilku układów z szyną SPI do jednej magistrali. Przebiegi czasowe wymagane do poprawnej pracy są przedstawione na rys. 5. Sygnał !SS jest wymagany tylko przy wyborze takiego trybu pracy. Maksymalna częstotliwość sygnału zegarowego wynosi około 100 kHz, ograniczenie to nie wynika z właściwości modułu MSSP, gdyż jest on w stanie odbierać dane z dużo większą prędkością, jednak przy szybszej transmisji mikrokontroler „nie zdąży” ich przetworzyć.

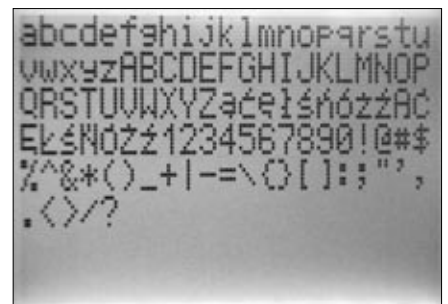
Komunikacja w standardzie I2C odbywa się również za pomocą modułu MSSP, który w tym celu należy zmodyfikować. Wykorzystywane są także te same rejestry robocze SSPSR i SSPBUF. Praca w trybie Slave jest w pełni kompatybilna ze standardem I2C i dlatego wyświetlacz może zostać dołączony do magistrali wraz z innymi układami, na przykład z pamięcią EEPROM. Adres, pod którym mikrokontroler będzie się zgłaszał na magistrali

I2C zależy od stanu przełącznika SW1 i może przyjąć jedną z czterech wartości: 90h, 92h, 94h, 96h. Wszystkie sygnały zgodne ze specyfikacją magistrali I2C są wykrywane i generowane przez sterownik, dzięki czemu jednostka centralna jest „powiadamiana” przerwaniem dopiero w momencie odebrania bajta danych, co może nastąpić tylko wtedy, gdy adres układu podany na magistrali jest zgodny z ustalonym przełącznikiem SW1. Dla tego interfejsu również należy zachować ograniczenie maksymalnej częstotliwości sygnału zegarowego do wartości około 100 kHz.

Pomimo przeniesienia wszystkich funkcji transmisji danych na sterowniki sprzętowe podczas odbioru strumienia danych występuje problem jednoczesnej obsługi wyświetlacza i analizowania poszczególnych bajtów. I tak wydanie kolejno kilku poleceń może spowodować, że pierwsze zostanie wykonane, a kolejne zostaną zignorowane lub nastąpi przekłamanie, gdyż procesor zamiast analizować te dane będzie zajęty obsługą wyświetlacza. Aby zapobiec takiej sytuacji w pamięci RAM mikrokontrolera została utworzona pamięć buforująca dane odbierane z modułów transmisji szeregowej. Bufor tworzy pamięć typu FIFO o pojemności 256 B. Działanie pamięci FIFO polega na tym, że zapisywane są do niej dane pod kolejnymi adresami przez jeden moduł (na przykład strumień danych z portu szeregowego, których nie można w czasie rzeczywistym analizować), a po zakończeniu pakietu lub równoległe inny moduł może te dane odczytywać i analizować ze znacznie mniejszą częstotliwością. Ideą pamięci FIFO jest to fakt, że niezależnie od ilości bajtów do niej zapisanych (liczba musi być mniejsza od całkowitej pojemności) kolejność zapisywanych i odczytywanych



Rys. 7. Przykład połączenia tekstu i grafiki



Rys. 8. Znaki ASCII wyświetlane przez wyświetlacz

**Tab.3 Zestawienie wszystkich komend realizowanych przez sterownik wyświetlacza**

Rodzaj komendy	Wydana komenda
Wyświetlanie znaków ASCII	'A' x y r k o d1 d2 ... 0 A – tryb wyświetlania znaków x- współrzędne na osi x (0...127) y- współrzędne na osi y (0...63) r- rozmiar czcionki. 0-5x7, 1-10x14 k- kolor. 0-wyłączony, 1-włączony, 2-zmiana stanu na przeciwny o- odświeżenie stanu wyświetlacza - (0,1) 0- bez odświeżenia, 1-z odświeżeniem d1,d2, ... – znaki ASCII 0 – znacznik końca podawania znaków Przykład: wyświetla napis ABC w lewym górnym rogu wyświetlacza 'A' 0 0 1 1 1 'A' 'B' 'C' 0
Rysowanie koła/okręgu	'K' x y p w k o x- współrzędne na osi x - (0...127) y- współrzędne na osi y - (0...63) p- promień koła/okręgu w- wypełnienie. 0 - bez wypełnienia -(okrąg), 1 - koło wypełnione k- kolor. 0-wyłączony, 1-włączony o- odświeżenie stanu wyświetlacza - (0,1) 0- bez odświeżenia, 1-z odświeżeniem Przykład: koło o promieniu 20piksli wypełnione, umieszczone na środku wyświetlacza 'K' 63 32 20 1 1 1
Rysowanie kwadratu, prostokąta, wypełnienie podanego obszaru kolorem, rysowanie ramki	'F' x1 y1 x2 y2 w k o x1- współrzędne początku figury na osi x - (0...127) x1- współrzędne początku figury na osi y - (0...63) x2- współrzędne końca figury na osi x - (0...127) y2- współrzędne końca figury na osi y - (0...63) w- wypełnienie. 0-wewnątrz obszar nie wypełniony - (ramka), 1 - wewnętrzny obszar wypełniony k- kolor. 0-wyłączony, 1-włączony, 2-zmiana stanu na przeciwny o- odświeżenie stanu wyświetlacza - (0,1) 0 - bez odświeżenia, 1-z odświeżeniem Przykład: prostokąt o 100x50 pikseli od lewego górnego rogu 'F' 0 0 100 50 1 1 1
Rysowanie linii	'L' x1 y1 x2 y2 k o x1- współrzędne początku linii na osi x - (0...127) x1- współrzędne początku linii na osi y - (0...63) x2- współrzędne końca linii na osi x - (0...127) y2- współrzędne końca linii na osi y - (0...63) k- kolor. 0-wyłączony, 1-włączony, 2-zmiana stanu na przeciwny o- odświeżenie stanu wyświetlacza - (0,1) 0 - bez odświeżenia, 1-z odświeżeniem Przykład: linia przebiegająca od lewego górnego, do prawego dolnego rogu wyświetlacza 'L' 0 0 127 63 1 1
Włączenie/wyłączenie pojedynczego piksela	'P' x y k o x- współrzędne na osi x - (0...127) y- współrzędne na osi y - (0...63) k- kolor. 0-wyłączony, 1-włączony, 2-zmiana stanu na przeciwny o- odświeżenie stanu wyświetlacza - (0,1) 0 - bez odświeżenia, 1-z odświeżeniem Przykład: zmienia stan na przeciwny piksela o współrzędnych x=20, y=30 'P' 20 30 2 1
Odświeżenie zawartości wyświetlacza	'O' Przykład: zawartość pamięci obrazu procesora jest kopiowana do pamięci wyświetlacza 'O'
Kasowanie zawartości wyświetlacza	'C' k o k- kolor. 0-wyłączony, 1-włączony, 2-zmiana stanu na przeciwny o- odświeżenie stanu wyświetlacza - (0,1) 0 - bez odświeżenia, 1-z odświeżeniem Przykład: wyłączenie wszystkich pikseli 'C' 0 1 Przykład2: włączenie wszystkich pikseli 'C' 1 1 Przykład3: zmiana stanu wszystkich pikseli na przeciwny - (negacja) 'C' 2 1
Zapis obrazu do pamięci EEPROM	'W' n n – numer obrazu - (0...31) Przykład: zapis aktualnego obrazu LCD pod numerem 1 'W' 1
Odczyt obrazu z pamięci EPROM	'R' n o n – numer obrazu - (0...31) o- odświeżenie stanu wyświetlacza - (0,1) 0- bez odświeżenia, 1-z odświeżeniem Przykład: wyświetlenie obrazu zapisanego pod numerem 1 'R' 1 1
Ustawienie intensywności podświetlania	'D' i i – intensywność podświetlania - (0...9) Przykład1: włączenie podświetlania z maksymalną intensywnością 'D' 9 Przykład2: całkowite wyłączenie podświetlania 'D' 0
Przerwanie podawania komendy	0xFD (hex) – podanie tej wartości unieważnia podane wcześniej parametry i umożliwia rozpoczęcie nowej komendy bez realizacji poprzednio rozpoczętej Przykład: przerwanie funkcji rysowania koła i wydanie komendy włączenia podświetlania 'K' 63 32 20 (FDh) 'D' 9
Zdalny reset sterownika	0xFE (hex) – podanie tej wartości powoduje zerowanie procesora sterującego wyświetlaczem i rozpoczęcie pracy, tak jak po włączeniu zasilania

nych bajtów jest taka sama, czyli bajt zapisany jako pierwszy będzie odczytany również jako pierwszy, drugi jako drugi, itd. W sterowniku dane do pamięci zapisywane są w przerwaniu generowanym przez wybrany sterownik transmisji szeregowej, a odczytywane są w głównej pętli programu. Dzięki temu wysłanie do sterownika kilku poleceń nie spowoduje „zgubienia” żadnego z nich, gdyż bajty będą zapisywane do bufora w przerwaniu, które zawsze musi być obsługiwane, a procesor będzie je kolejno realizował w głównej pętli programu w „wolnym czasie”. W ten sposób jednorazowo można wysłać pakiet komend składający się 256 bajtów, bez obawy, że nie zostaną zrealizowane.

## Montaż

Sterownik wyświetlacza został zmontowany na płytce dwustronnej, której widok przedstawiony jest na **rys. 6**. Montaż należy rozpocząć od wlutowania rezystorów, następnie należy wlutować podstawki pod układy scalone. W dalszej kolejności należy zamontować tranzystor, kondensatory, przełącznik i na końcu złącza. Przy czym złącza CON1...CON3 montowane są od strony elementów, natomiast złącze pod wyświetlacz montowane jest od strony lutowania. Wyświetlacz posiada szesnaście wyprowadzeń służących do jego zasilania i sterowania oraz dwa dodatkowe służące do doprowadzenia napięcia zasilającego diody podświetlające wyświetlacz. Sygnały zasilania diod nie są wyprowadzone na złącze wyświetlacza i dlatego należy je połączyć przewodami do dodatkowych pól lutowniczych umieszczonych na płytce sterownika i oznaczonych odpowiednio: A-A (anoda), K-K (katoda).

Po zmontowaniu całego układu można przejść do obsługi wyświetlacza. Cały sterownik należy zasilac napięciem o wartości 5 V (150 mA).

## Obsługa

Przed rozpoczęciem sterowania wyświetlaczem należy ustalić rodzaj i parametry interfejsu poprzez odpowiednie ustawienie przełącznika SW1. Stan przełącznika oraz odpowiadające mu tryby pracy przedstawione są w **tab. 2**. Rodzaj komunikacji jest wybierany za pomocą przełączników S1 i

**Tab. 4. Wartości HEX przypisane polskim znakom diakrytycznym**

Znak	Wartość HEX	Kombinacja klawiszy
ą	0xB9	Alt+a
ć	0xE6	Alt+c
ę	0xEA	Alt+e
ł	0xB3	Alt+l
ń	0xF1	Alt+n
ś	0x9C	Alt+s
ó	0xF3	Alt+o
ż	0xBF	Alt+z
ź	0x9F	Alt+x
Ą	0xA5	Alt+A
Ć	0xC6	Alt+C
Ę	0xCA	Alt+E
Ł	0xA3	Alt+L
Ń	0xD1	Alt+N
Ś	0x8C	Alt+S
Ó	0xD3	Alt+O
Ż	0xAF	Alt+Z
Ź	0x8F	Alt+X

S2, przełączniki S3 i S4 natomiast służą do zmiany parametrów danego typu komunikacji. Dla pracy w trybie RS232 można wybrać w zależności od potrzeb jedną z czterech prędkości pracy. Dla pracy w trybie I2C przełączniki S3, S4 umożliwiają zmianę adresu, pod którym będzie się zgłaszał wyświetlacz na magistrali I2C. Trzeci tryb odnosi się do pracy w trybie SPI z aktywnym wejściem !SS, natomiast w czwartym trybie do pracy nie jest wymagany sygnał !SS. Dla komunikacji w trybie SPI przełączniki S3 i S4 nie są używane, dlatego ich stan nie ma znaczenia.

Niezależnie od wybranego interfejsu obsługa wyświetlacza przebiega tak samo i jest realizowana za pomocą komend z odpowiednimi parametrami. Każda komenda rozpoczyna się od podania odpowiedniego znaku ASCII, a następnie parametrów w postaci binarnej o zmiennej liczbie, w zależności od rodzaju komendy. Spis wszystkich komend znajduje się w **tab. 3**. Wydanie polecenia wyświetlania znaków ASCII jest tak zbudowane, że raz wydane powoduje wyświetlenie znaków do momentu pojawienia się bajta o wartości 00 (hex). Jeśli napis dojdzie do końca linii, to kolejny znak automatycznie zostanie przeniesiony na początek następnej linii. A po osiągnięciu ostatniej pozycji wyświetlacza (prawy dolny róg) następuje przeskoczenie na początek (lewy górny róg). Dodatkowo jeśli w ciągu znaków ASCII pojawi się wartość 0Ch, to zostanie ustawiony adres początku linii i kolejne znaki będą wpisywane od początku tej linii. Jeśli zaś pojawi się wartość 0Ah, to

nastąpi przejście do następnej linii. Podając wartość 0Ah i 0Ch kursor zostanie przeniesiony na początek następnej linii. W ten sposób bez wychodzenia z trybu znakowego można ustawić pozycje kursora. Ważnym parametrem jest ustawienie koloru wyświetlanego tekstu. Jeśli zostanie wybrany tryb wyłączenia pikseli i jeśli wcześniej wyświetlacz był wykasowany (wyłączone wszystkie piksele), to na wyświetlaczu nie pojawi się nic, ponieważ tekst i tło mają taki sam kolor. Jeśli jednak wszystkie lub część pikseli w miejscu, w którym ma się pojawić napis będą włączone, to napis powstanie z wygaszonych pikseli. Dla koloru znaków ustawionej w tryb włączania pikseli sytuacja będzie odwrotna. Uniwersalną funkcję pełni trzeci tryb koloru - zmiana stanu na przeciwny. Zastosowanie tego trybu spowoduje, że jeśli tło napisu będzie miało włączone piksele, a jeśli tło jest utworzone z wyłączonych pikseli, to napis powstanie z wyłączonych. W ten sposób niezależnie od tła napis powstanie. Zależność ta odnosi się także do funkcji graficznych: rysowania pola, linii, włączenia piksela.

Również funkcja kasowania wyświetlacza działa na tej zasadzie, dzięki czemu podając jako parametr wartość 2 cała pamięć wyświetlacza zostanie zanegowana i to co było włączone, będzie wyłączone, a to co było wyłączone - będzie włączone.

Przykład wyświetlania w trybie tekstowym i graficznym jest przedstawiony na **rys. 7**. W pamięci procesora zdefiniowane są także polskie znaki diakrytyczne, których wyświetlenie następuje po podaniu odpowiedniej wartości zgodnej z wartościami generowanymi przez klawiaturę komputera. Spis wszystkich znaków i odpowiadające im wartości znajdują się **tab. 4**, natomiast wygląd czcionek dla wszystkich znaków wyświetlanych przez wyświetlacz przedstawia **rys. 8**.

Wspólnym dla wszystkich funkcji wyświetlania jest parametr odświeżania wyświetlacza, jeśli ma wartość równą zero, to pomimo wysłania polecenia, na przykład narysowania kwadratu, figura ta nie pojawi się na wyświetlaczu, a zostanie zapisana tylko w pamięci procesora. W

ten sposób można stworzyć cały obraz zawierający różne elementy, a następnie całość wyświetlić jednocześnie poprzez wydanie polecenia odświeżenia wyświetlacza.

Funkcja zapisu obrazu do pamięci EEPROM umożliwia zachowanie, w celu późniejszego odtworzenia, zawartości pamięci wyświetlacza. Aby zapisać obraz w tej pamięci należy go wcześniej wyświetlić na wyświetlaczu lub tylko zapisać w pamięci procesora. Wydanie polecenia zapisu do pamięci EEPROM spowoduje skopiowanie całej zawartości pamięci pośredniej zawartej w procesorze do pamięci EEPROM pod wskazanym numerem. Ze względu na typ zastosowanej pamięci po wydaniu polecenia zapisu należy odczekać czas 100ms przed wydaniem następnego polecenia do sterownika.

Ostatnim poleceniem jest polecenie całkowitego resetu procesora sterującego. Polecenie to posiada najwyższy priorytet i dlatego niezależnie od tego czy w danej chwili realizowane są inne czynności, wysłanie wartości FE (hex) spowoduje ponowny start procesora, tak jak to ma miejsce przy włączeniu zasilania.

**Krzysztof Pławiuk, EP**  
krzysztof.plawiuk@ep.com.pl

#### SPIS ELEMENTÓW

##### Rezystory

R1: 4,7kΩ  
R2...R5: 2kΩ  
R6: 1kΩ  
R7: 4,7Ω  
R8, R9: 4,7kΩ  
PR: potencjometr 10kΩ

##### Kondensatory

C1, C2: 30pF  
C3: 100nF  
C4...C8: 10μF/16V

##### Półprzewodniki

T1: BD139  
US1: PIC18LF252 zaprogramowany  
US2, US3: ICL7660  
US4: AT24C256

##### Inne

CON1: Goldpin 1x6 męski  
CON2, CON3: Goldpin 1x4 męski  
SW1: przełącznik DIP4  
X1: rezonator kwarcowy 10MHz  
Wyświetlacz graficzny JM12864A  
Podstawka DIP8 - 3szt.  
Podstawka DIP28 (300mils) - 1szt.

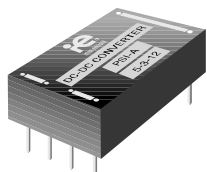


# EBS

Ink Jet Systems

Renomowany producent przemysłowych drukarek INK-JET oferuje wysokiej klasy elementy automatyki:

**miniaturowe przetwornice DC/DC**  
do zastosowań w obwodach zasilania układów cyfrowych i analogowych



napięcie wyjściowe pojedyncze lub podwójne  
galwaniczna separacja wejście - wyjście  
galwaniczna separacja wyjść  
współpraca przetwornic szeregowo lub równoległa  
odporne na zwarcie

**aktywny detektor podczzerwieni**  
do zastosowań w układach automatyki i zabezpieczeń

małe wymiary budowy (M18x1)  
duża odporność na zakłócenia  
wbudowany wskaźnik zadziałania  
wyjście odporne na zwarcie  
wykonania PNP, NPN



**EBS**  
Ink Jet Systems

EBS Ink-Jet Systems Poland Sp. z o.o.

ul. Tarnogajska 11/13  
50-512 Wrocław  
tel. (0-71) 367 04 11  
fax (0-71) 373 32 69

**GURU**  
CONTROL SYSTEMS

ul. Stomińskiego 1, 00-204 Warszawa  
tel./fax.: (0-22) 831-10-42, GSM: 0-602 262 230  
e-mail: info@guru.com.pl [www.guru.com.pl](http://www.guru.com.pl)

## KOMPUTERY PRZEMYSŁOWE (IPC)



- monitory LCD, ekrany dotykowe
- klawiatury pyłoszczelne, zasilacze
- obudowy przemysłowe, panele operatorskie, magistrale pasywne
- komputery jednopłytkowe (SBC)
- karty ISA i PCI we-wy analogowych i cyfrowych, mikro-moduły PC/104
- adaptory i karty wieloportowe RS-232/422/485, karty IEEE-488

## Inteligentne moduły pomiarowo-sterujące I-7000

- we-wy cyfrowe, analogowe, licznikowe, przekaźnikowe, termoparowe, RTD; łatwe programowanie rozkazami ASCII
- łączenie w sieci do 2048 modułów (256 bez repeatera)
- komunikacja 2-przewodowa RS-485 do 115200 bps, obsługa całej sieci modułów jednym portem RS-232 z komputera PC lub modułu mikrokomputera I-7188, montaż na szynie DIN
- różne prędkości transmisji i formaty danych w jednej sieci
- zasilanie od 10 do 30V DC, izolacja 3000V, watch-dog lokalny, watch-dog sieciowy, stan bezpieczny wyjść
- wyświetlacz LCD z klawiaturą, terminale dotykowe, radiomodemy, zasilacze, obudowy, oprogramowanie



## EUROCARD - profesjonalny standard przemysłowy:



- niezawodne połączenie z magistralą typu gniazdo-wtyk 96 stykowy
- karty zamocowane wzdłuż 4 boków
- znakomite chłodzenie - swobodny przepływ powietrza z dołu do góry
- wygodny serwis i rekonfiguracja - wymiana kart i zasilaczy bez wyjmowania obudowy z szafy i jej rozkręcania
- dostęp do wszystkich złączy obiektowych, przelączników, sygnalizatorów - od przodu

## Oferujemy w standardzie EURO:

- ✓ komputery IPC 486, sterowniki
- ✓ karty procesorów i komputerów jednopłytkowych 80C52, 80C186, 486
- ✓ izolowane karty: we-wy cyfrowych, analogowych, liczników rewersyjnych,
- ✓ wielokanałowe interfejsy pętli prądowej
- ✓ koncentratory interfejsów pętli prądowej,
- ✓ kasety, magistrale, zasilacze, adaptory sygnałów analogowych
- ✓ projekty i dostawy kart w/g wymagań klienta



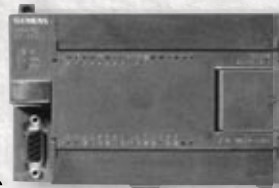
projektowanie, wdrożenia, produkcja



## AUTOMATYKA PRZEMYSŁOWA I ELEKTROTECHNIKA KOLEJOWA

- bezstykowe czujniki zbliżeniowe
- tachometry
- liczniki impulsów i czasu
- wskaźniki temperatury
- wskaźniki prądu i napięcia
- układy kontroli ruchu
- zasilacze przemysłowe 24VDC
- przekaźniki czasowe
- styczniki AC i DC
- złącza przemysłowe
- przetłączniki i inne elementy stykowe
- sterowniki SIMATIC S7-200, S7-300
- falowniki SINAMICS, MICROMASTER
- panele operatorskie SIMATIC HMI
- moduły logiczne LOGO!
- przetworniki obrotowo-impulsowe

**Warsztaty z zakresu SIMATIC S7-200**



Więcej szczegółowych informacji:

**IMPOL-1 Sp.j.**  
02-255 Warszawa  
ul. Krakowiaków 103  
tel. (22) 886-56-02  
fax (22) 886-56-04  
[www.impol-1.pl](http://www.impol-1.pl)

**Na żądanie wysyłamy bezpłatne katalogi w/w wyrobów**