

Sterownik VGA

AVT-539



Prezentowany sterownik VGA jest uniwersalnym urządzeniem umożliwiającym wizualizację znaków alfanumerycznych na monitorach VGA.

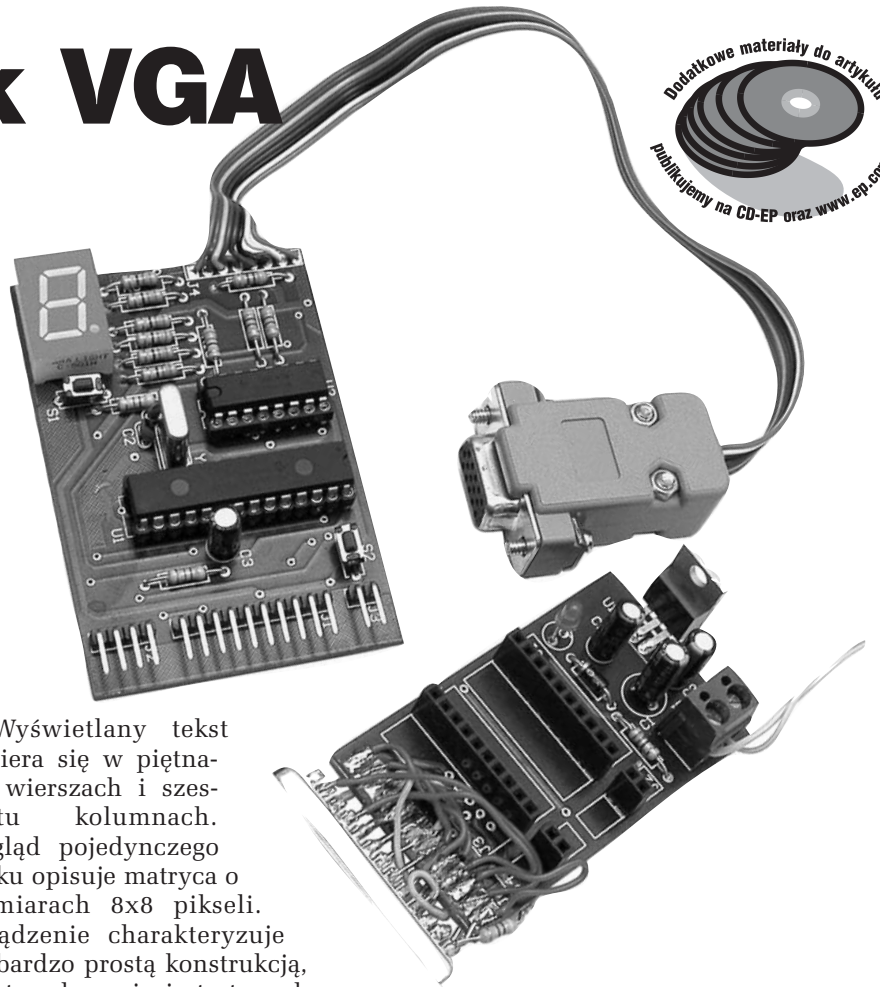
Rekomendacje: interesujące opracowanie ilustrujące sposób tworzenia sygnału wideo dla monitorów VGA, charakteryzujące się przy tym sporymi walorami praktycznymi - urządzenie można bowiem wykorzystać jako programowalny sterownik VGA.

Wyświetlany tekst zawiera się w piętnastu wierszach i szesnastu kolumnach. Wygląd pojedynczego znaku opisuje matryca o rozmiarach 8x8 pikseli. Urządzenie charakteryzuje się bardzo prostą konstrukcją, koszt wykonania jest stosunkowo niski, dzięki czemu może stanowić (oczywiście w pewnych aplikacjach) alternatywę dla wyświetlaczy LCD. Prosty, a zarazem uniwersalny interfejs komunikacyjny oparty o interfejs SPI umożliwia łatwe wykorzystanie modułu w różnorodnych układach mikroprocesorowych, a szczególnie w urządzeniach pomiarowych.

Do wysterowania monitora trzeba wytworzyć co najmniej pięć sygnałów: synchronizacji poziomej i pionowej oraz trzy sygnały wizyjne składowych koloru, którym to na typowo stosowanym złączu DB15 karty VGA odpowiadają styki 13 (H-SYNC), 14 (V-SYNC) oraz dla informacji wizyjnej: 1 (R), 2 (G), 3 (B). Pierwszy z nich H-SYNC służy synchronizacji wiązki elektronów na monitorze do początku kolejnej wyświetlanej linii. Drugi przenosi wiązkę na początek kolejnego ekranu. Oba te sygnały muszą spełniać pewne kryteria czasowe, które opisano w **tab. 1**.

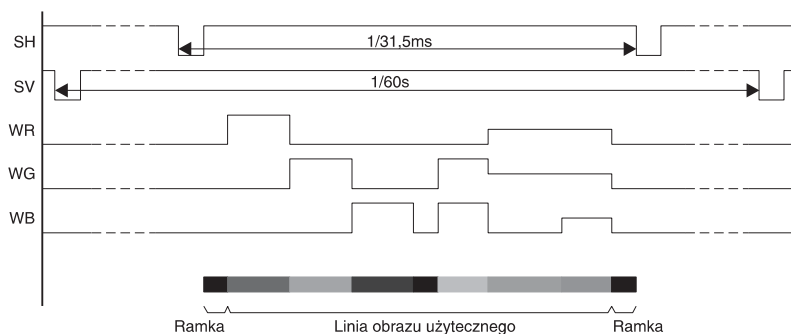
Pomiędzy kolejnymi impulsami synchronizacji poziomej należy odpowiednio sterować wejścia RGB w celu uzyskania na ekranie monitora odpowiedniego obrazu. Sygnały podawane na wejścia RGB niosą (analogowo) informację o wyświetlanym obrazie. Przykładowy przebieg czasowy na liniach sterujących monitorem VGA pokazano na **rys. 1**.

W prezentowanym układzie do generacji sygnałów sterujących użyto mikrokontrolera, który spełnia rów-

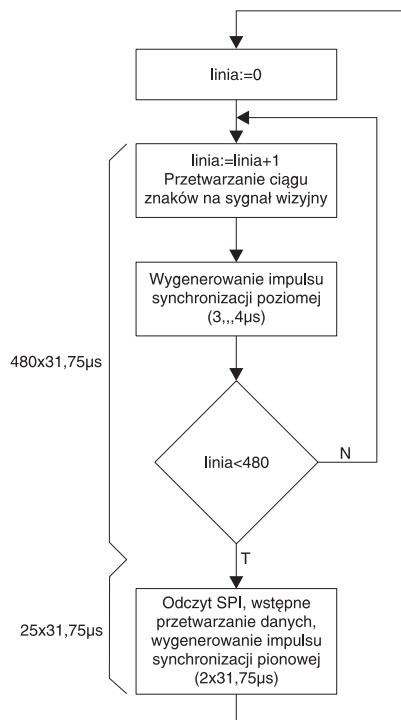


Tab. 1. Parametry czasowe dla wybranych trybów pracy VGA

Rozdzielczość	Szerokość pasma [MHz]	Częstotliwość pozioma [kHz]	Częstotliwość pionowa [Hz]
640x400	25,175	31,50	70
720x400	28,322	31,50	70
640x480	25,175	31,50	60
640x350	25,175	31,50	70



Rys. 1. Przebiegi czasowe podczas wyświetlania obrazu w trybie VGA (640 na 480 punktów)



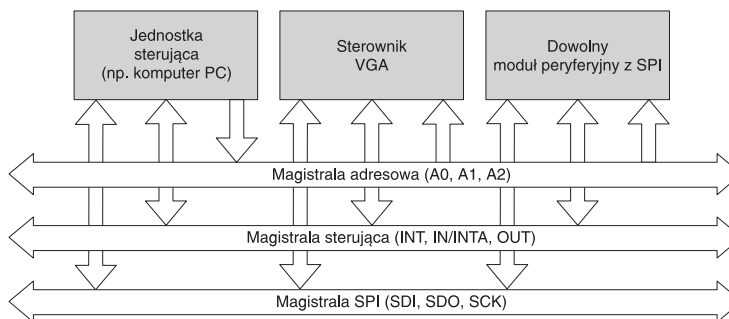
Rys. 2. Algorytm działania mikrokontrolera

niez rolę inteligentnego interfejsu zapewniającego komunikację z otoczeniem za pomocą interfejsu SPI. Realizacja większości funkcji odbywa się na drodze programowej. Szkic algorytmu działania mikrokontrolera przedstawiono na **rys. 2**.

Tak przedstawiony algorytm wygląda nieskomplikowanie, jednak rygorystyczne zależności czasowe stawiane sygnałom synchronizacji były dużym wyzwaniem podczas tworzenia oprogramowania.

Do wytworzenia przebiegów z danymi o obrazie jest konieczny przetwornik cyfrowo-analogowy. Sterowanie takim przetwornikiem może okazać się dość złożone i czasochłonne. Z tych względów zdecydowano się na zastosowanie jednobitowej konwersji C/A, którą można łatwo zrealizować używając do tego jednej linii portu mikrokontrolera. Rozwiązanie takie jest oszczędne układowo i zmniejsza liczbę komórek pamięci potrzebnych do przechowywania obrazu. Niestety niesie to również ograniczenie co do liczby możliwych do wyświetlenia kolorów (odcieni), która wynosi tylko dwa.

Generacja sygnału wizyjnego w prezentowanym urządzeniu jest wspomagana sprzętowo poprzez zastosowanie bloku do transmisji szeregowej USART i jego synchronicznego trybu pracy, będącego jednym z



Rys. 3. Schemat blokowy typowej aplikacji korzystającej ze sterownika VGA

wewnętrznych urządzeń zastosowanego mikrokontrolera. Takie niekonwencjonalne rozwiązanie w znacznym stopniu uprościło program i przyczyniło się do uzyskania lepszej rozdzielczości. Niekorzystnym efektem ubocznym jest trudny do wyeliminowania biały pionowy pasek po lewej stronie ekranu obecny podczas pracy układu. Mając na celu minimalizację liczby transmitowanych danych oraz minimalizację zapotrzebowania na pamięć danych, wyświetlane obrazy mogą składać się jedynie ze znaków alfanumerycznych, których wzorce (kształty znaków) zostały zaszyte na stałe w pamięci programu. Rozdzielczość została dobrana na drodze eksperymentalnej. W wyniku wielu prób z różnymi wersjami oprogramowania udało się w efekcie uzyskać następujące parametry: 16 kolumn na 15 wierszy przy matrycy znaku 8x8. Kody ASCII przesyłane do sterownika muszą zawierać się w domkniętych przedziałach od 0x30 (spacja) do 0x3E ('>') oraz od 0x41 ('A') do 0x5F ('_'). Kody z poza tych zakresów są mogą powodować błędy w funkcjonowaniu modułu. Poza tymi wartościami dopuszczalne jest również przesyłanie specjalnych kodów sterujących, pozwalających dokonywać operacji takich jak czyszczenie ekranu i ustawianie kursora. Zestawienie wszystkich dostępnych kodów przedstawiono w **tab. 2**.

Proces transmisji danych jest oparty o dobrze znany interfejs SPI zmodyfikowany o możliwość przypisania adresu logicznego, pozwalający na pracę w konfiguracji magistralowej. Na **rys. 3** przedstawiono schemat blokowy typowego systemu cyfrowego przystosowanego do współpracy ze sterownikiem VGA.

Na schemacie wyróżniono trzy podstawowe magistrale:

- adresową,
- sterującą,
- SPI.

Magistrala adresowa składa się z trzech linii (A0...A2) służących do przesyłania informacji o adresie aktualnie wybranego sterownika. Linie te są dwustanowe, jednokierunkowe. Liczba zastosowanych linii pozwala zaadresować do 8 sterowników dołączonych do jednej magistrali.

Magistrala sterująca składa się z linii żądania przerwania INT, linii potwierdzeń IN/INTA i linii wyjściowej ogólnego przeznaczenia OUT. Linia INT jest jednokierunkowa (wyjście), dwustanowa, przy czym aktywny jest stan niski. Dzięki takiej konstrukcji nie ma konfliktów na magistrali, w przypadku gdy do magistrali jest dołączony więcej niż jeden sterownik VGA lub inny układ z interfejsem SPI. Żądanie obsługi jest zgłaszane do systemu sterującego za pomocą stanu niskiego na linii INT. Linia

Tab. 2. Kody sterujące pracą sterownika VGA

Kod	Opis
0x19	Przesunięcie kursora o jedną pozycję w prawo
0x1A	Przesunięcie kursora o jedną pozycję w lewo
0x1B	Przesunięcie kursora o jedną pozycję w górę
0x1C	Przesunięcie kursora o jedną pozycję w dół
<0x01; 0x18>	Wykasowanie 1/24 ekranu, wartość z przedziału określa kasowany fragment ekranu, np. 0x01 oznacza skasowanie pierwszego i połowy drugiego wiersza, 0x18 to skasowanie ostatniego i od połowy przedostatniego wiersza
<0x80 ; 0xF8>	Ustawienie kursora na pozycję („wartość z przedziału” - 0x80)/2

INTA służy do transmisji sygnału potwierdzenia przerwania od modułu, może być także użyta jako linia wejściowa ogólnego przeznaczenia. Aktywnym stanem INTA jest stan niski (gdy nieaktywny to wysoka impedancja).

Ostatnią z linii sterujących jest OUT. Jej funkcja nie jest ściśle zdefiniowana i może służyć np. do potwierdzenia odbioru przez jednostkę centralną. W przypadku modułu VGA pozostaje ona niewykorzystywana.

W przypadku modułu VGA przesył danych nie może odbywać się w dowolnym czasie. Konstrukcja programu pozwala na odbiór jedynie w trakcie przedziału wygaszania pionowego (czarna ramka u góry i u dołu obrazu). Moduł swoją gotowość do transmisji sygnalizuje poprzez wystawienie sygnału niskiego na linii INT. Urządzenie obsługujące taki moduł, chcąc wyprowadzić daną, musi w odpowiednim czasie (mniejszym niż 30 μ s) odpowiedzieć na taki sygnał, ustawiając na liniach A0, A1, A2 adres modułu. Prawidłowy adres powoduje ustawienie przez moduł VGA sygnału niskiego na INTA i powrót do stanu spoczynkowego (wysoka impedancja) linii INT w czasie nie dłuższym niż 6 μ s. Po takiej sekwencji może nastąpić nadawanie.

WYKAZ ELEMENTÓW

Rezystory

R1..R3: 470 Ω
R4, R12: 47 k Ω
R5..R11: 330 Ω
R13..R15: 4,7 k Ω
R16: 10 k Ω
R17: 220 Ω

Kondensatory

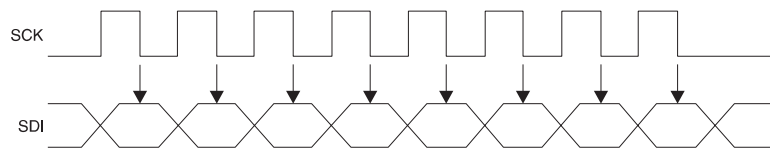
C2: 15 pF
C3..C5: 1 μ F/25V
C6: 47 μ F/25V

Półprzewodniki

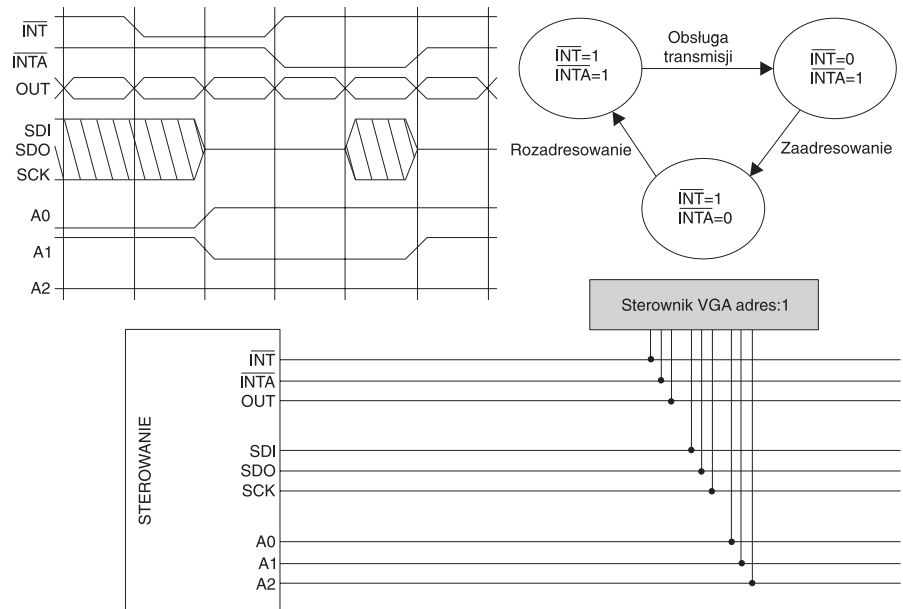
D1: 1N4007
D2: LED zielona
Q1: BC337
U1: PIC16F876-DIP28
U2: 4511
U3: 7805

Różne

Y1: rezonator kwarcowy 20 MHz
S1, S2: mikrowłazcznik
DS1: wyświetlacz 7-segmentowy (wspólna katoda)
J1: goldpin 1x8
J2: goldpin 1x4
J3, J5, J9, J10: goldpin 1x2
J4: goldpin 1x8
J6: DB25 (żeński)
J7, J8: goldpin 1x10



Rys. 4. Dane na linii SDI są zatrzymywane opadającym zboczem sygnału zegarowego



Rys. 5. Sygnały na magistrali podczas transmisji do sterownika VGA

Moduł VGA jest interfejsem SPI typu Slave. Stanem spoczynkowym zegara SCK jest stan niski. Próbkowanie danych na linii SDI następuje podczas narastającego zbocza SCK. Na rys. 4 przedstawiono przebiegi czasowe sygnałów SDI i SCK na magistrali sterującej podczas transmisji jednego bajtu.

Liczba transmitowanych bajtów jest ograniczona przez maksymalny całkowity czas przeznaczony na transmisję, określany od momentu ustawienia INT w stan niski. Wynosi on 254 μ s. Dodatkowo kolejne bajty nie mogą przychodzić częściej niż co 3,5 μ s. Wszystkie te ograniczenia wynikają z uproszczonej konstrukcji sprzętowej modułu. Koniec transmisji, a co za tym idzie zwolnienie magistrali, może nastąpić wcześniej niż czas określony wyżej jako maksymalny czas transmisji. Sygnałem dla modułu VGA świadczącym o zakończeniu przesyłu jest ztheadresowanie (linie A0...A2 przyjmują stany nie odpowiadające adresowi przypisanemu do modułu), w wyniku którego wszystkie linie ustawiane są w stan wysokiej impedancji. Na rys. 5 przedstawiono przebiegi czasowe sygnałów

na magistrali podczas kolejnych faz transmisji danych.

W sterowniku zastosowano mikrokontroler PIC16F876-20 oznaczony na schemacie elektrycznym (rys. 6) symbolem U1. Jego zadaniem jest odczyt danych z interfejsu komunikacyjnego, ich interpretacja oraz sterowanie sygnałami wizji i synchronizacji VGA. Sterowanie sygnałem wizyjnym odbywa się z wykorzystaniem linii RC7, będącej wyjściem DT dla bloku USART wbudowanego w PICa. Rezystory R1, R2 i R3 służą do uzyskania odpowiedniej barwy i jasności. Ich wartości dobrano eksperymentalnie tak, aby uzyskać właściwy odcień bieli. Linie 2 i 5 portu B (wyprowadzenia 23 i 26 układu U1) obsługują odpowiednio: synchronizację pionową i poziomą. Wyświetlacz DS1, układ U2 i linie RA0...RA3 odpowiadają za prezentację adresu przypisanego modułowi i wykorzystywanego podczas komunikacji. Mikroprzełącznik S1 wraz z rezystorem R12 i linią RB0 pozwalają na zmianę tego adresu. Dodatkowo na złączu J2 wyprowadzono linie portów RB3, RB6 i RB7 odpowiedzialne za proces programowania i debugo-

wania w systemie, tzw. *In-Circuit Serial Programming* i *In-Circuit Debugging*. Mogą one posłużyć do eksperymentów z *firmware* sterownika. Pozostałe wykorzystywane wyprowadzenia mikrokontrolera odpowiadają za komunikację.

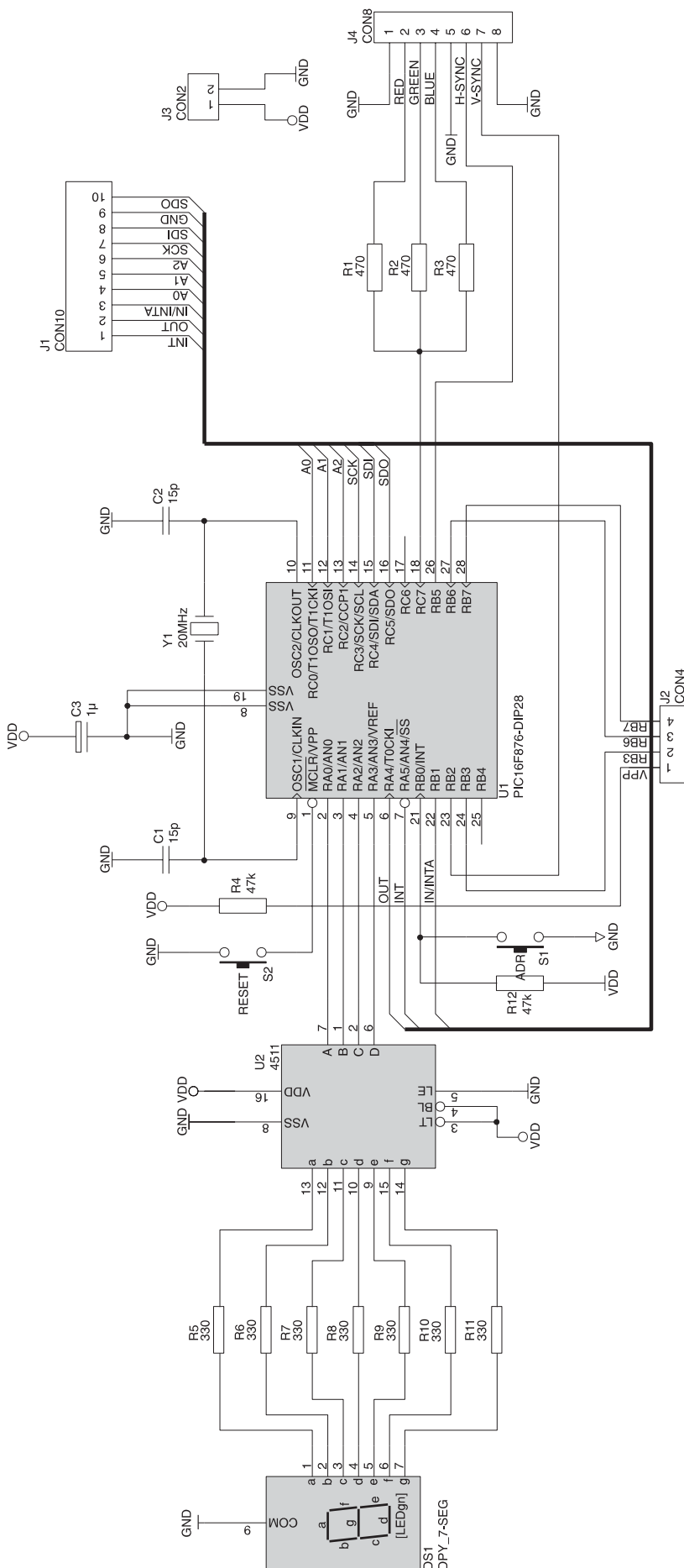
Montaż i uruchomienie

Montaż układu jest niezwykle prosty (schemat montażowy płytki pokazano na rys. 7). Jako złącza J2, J3, J4 wykorzystano końcówki typu PB-X (męskie). Dzięki temu możliwe jest wygodne mechaniczne zamocowanie modułu w docelowym środowisku pracy. Do J4 przylutowano kabel z gniazdem służącym do podłączenia monitora. Jego budowę przedstawiono na rys. 8.

Próbnę uruchomienie układu może odbyć się bez jakichkolwiek dodatkowych elementów sterujących. W tym celu złącze J3 należy zasilić napięciem 5 V, a kabel z gniazdem DB15 podłączyć do monitora VGA. Przy prawidłowo zmontowanym układzie na ekranie powinna widnieć plansza tytułowa zawierająca nazwę modułu oraz imię i nazwisko autora. Być może konieczna będzie jedynie korekta ustawień monitora decydująca o położeniu obrazu i jego jasności. Dodatkowo można wypróbować działanie wskaźnika aktualnie przypisanego adresu oraz dokonać jego konfiguracji. W trakcie normalnej pracy jest on zgaszony. Naciśnięcie S1 w

Tab. 3. Przyporządkowanie linii magistrali SPI do wyprowadzeń portu LPT

Styk LPT	Sygnal	Moduł VGA
1	~Strobe	Nie używane
2	D0	OUT
3	D1	Nie używane
4	D2	SDO
5	D3	SCK
6	D4	Nie używane
7	D5	A0
8	D6	A1
9	D7	A2
10	~ACK	~INT
11	Busy	Nie używane
12	PE	Nie używane
13	Selected	IN/INTA
14	~AutoLF	Nie używane
15	~Error	SDI
16	~INIT	Nie używane
17	~SLCT_IN	Nie używane
18-25	GND	GND

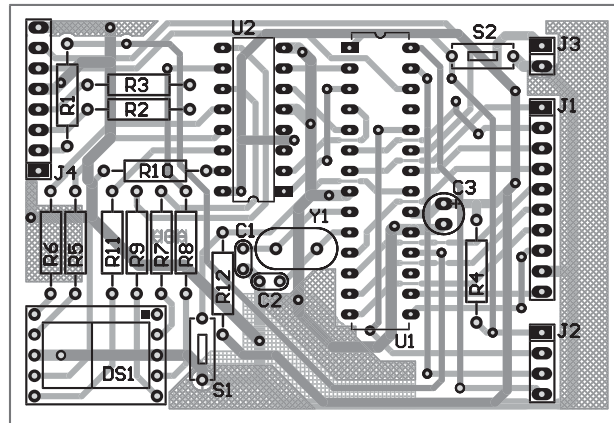


Rys. 6. Schemat elektryczny sterownika VGA

tym stanie powoduje pojawienie się na wyświetlaczu siedmiosegmentowym adresu aktualnie skojarzonego z modulem. Kolejne załączenie S1 w trakcie świecenia LED powoduje zmianę adresu na kolejny. Adres jest zapisywany do wewnętrznej pamięci EEPROM mikrokontrolera, dzięki czemu jest on pamiętany również po wyłączeniu zasilania. Diody wyświetlacza po czasie około 2 s od czasu ostatniego naciśnięcia S1 samoczynnie gasną.

Dla celów demonstracyjnych opracowano specjalny układ umożliwiający sterowanie modulem VGA z poziomu komputera PC. Schemat tego interfejsu przedstawiono na rys. 9. Za jego pomocą poprzez programowanie linii portu drukarki przesyłać kody wyświetlanych znaków do modułu VGA.

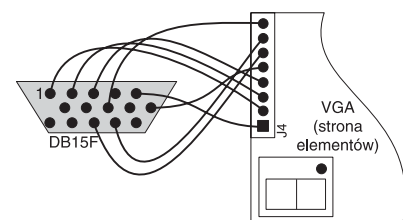
Budowa interfejsu jest bardzo prosta i w głównej mierze ogranicza się do odpowiedniego połączenia linii portu LPT z liniami magistrali modułu VGA. Wyjątkiem są linie INT i INTA, które są polaryzowane dodatnio za pośrednictwem rezystorów R4 i R5, aby wymusić na nich spoczynkowo stan wysoki (jedynka pasywna). Oprócz tego linia INT jest zanegowana na Q1. Jest to konieczne w przypadku gdy miałyby być wykorzystane przerwanie drukarki generowane podczas zmiany stanu niskiego na wysoki nACK, tj. odwrotnie niż dla



Rys. 7. Schemat montażowy płytki sterownika

magistrali zaimplementowanej w module VGA. W urządzeniu przewidziano możliwość jednoczesnej współpracy z dwoma modułami opartymi na omawianej magistrali. Do ich instalacji służą gniazda typu PB-X na schemacie oznaczone jako J1, J2, J3 i J4. Dodatkowo układ zawiera stabilizator napięcia służący do zasilania podpinanych modułów, zbudowany w oparciu o kostkę 7805. W tab. 3 zawarto przyporządkowanie linii magistrali do wyprowadzeń portu LPT.

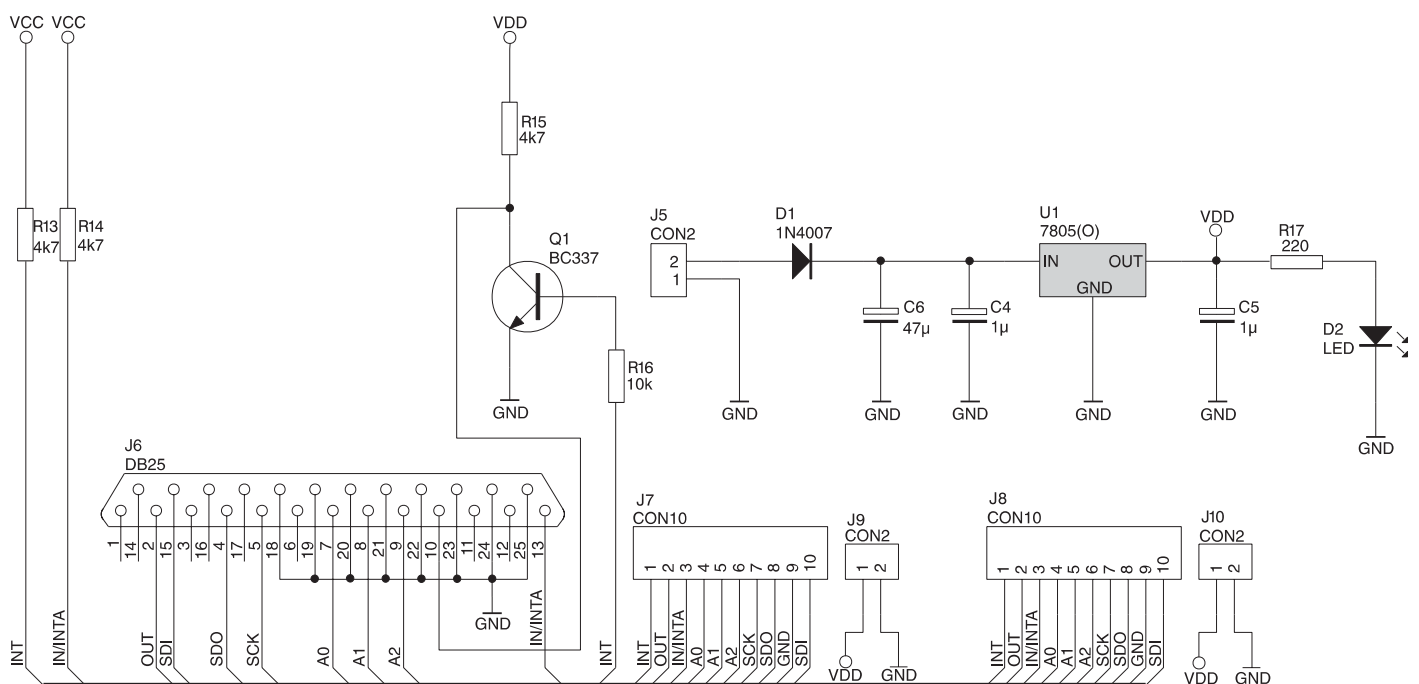
Przykładowy program napisany w C dla systemu DOS (FreeDOS, MS-DOS) - *lpmain* - czyta znaki z klawiatury i przesyła je do modułu o adresie 0, a do modułów o adresach 1, 2, 3 wysyła ciągiem kody ASCII o warto-



Rys. 8. Budowa kabla łączącego sterownik z monitorem VGA

ściach odpowiednio '1', '2' i '3'. Jego wywołanie wymaga podania parametru *-p* z argumentem określającym adres portu, np. *lpmain -p 0x378*. W wyznaczeniu odpowiedniego adresu pomocna jest opcja *-t*, wyszukująca dostępne pory równoległe.

Marcin Orłowski



Rys. 9. Schemat interfejsu sterującego modulem VGA