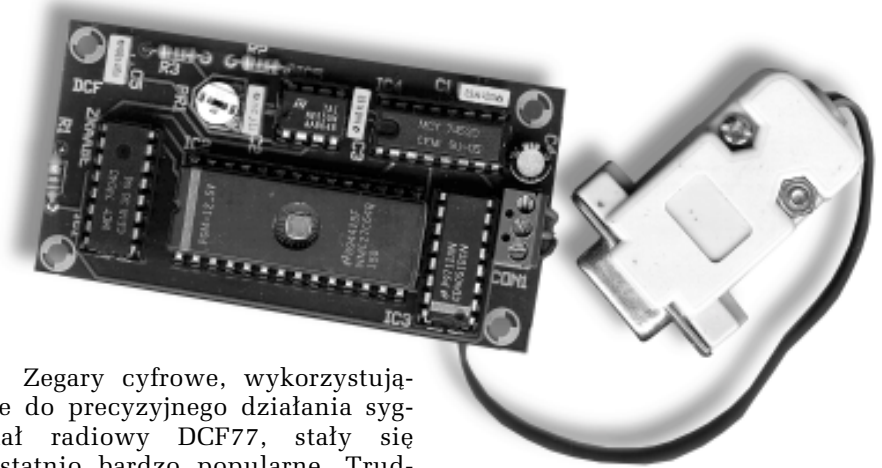


Emulator odbiornika DCF77

kit AVT-423



Doświadczenia nasze i Czytelników wskazują, że urządzenie emulujące nadajnik sygnału DCF-77 jest bardzo potrzebne w pracowniach wszystkich konstruktorów wykorzystujących w swoich opracowaniach ten popularny wzorzec czasu.

Prosta konstrukcja emulatora, łatwość konfiguracji i jego dobre cechy użytkowe powodują, że od tej pory uruchamianie zegarów synchronizowanych przez wzorzec przestanie stanowić problem.

Zegary cyfrowe, wykorzystujące do precyzyjnego działania sygnał radiowy DCF77, stały się ostatnio bardzo popularne. Trudno się temu dziwić, jest to bowiem najprostsza i najtańsza metoda zbudowania zegara o doskonałych, a nawet praktycznie nieosiągalnych innymi metodami parametrach. Wykorzystywanie sygnału DCF77 pozwala bowiem zbudować zegar o dokładności wynoszącej 1 sekundę na 5 milionów lat! Nie sądzę, aby komukolwiek z nas chciało się czekać tyle czasu w celu empirycznego stwierdzenia tego sekundowego odchylenia!

Zbudowanie zegara „napędzanego” sygnałem nadawanym z Mainflingen w Niemczech nie jest trudne i w najprostszym przypadku zegar taki będzie się składał głównie z odpowiednio zaprogramowanego procesora, wyświetlaczy i kilku elementów dodatkowych.

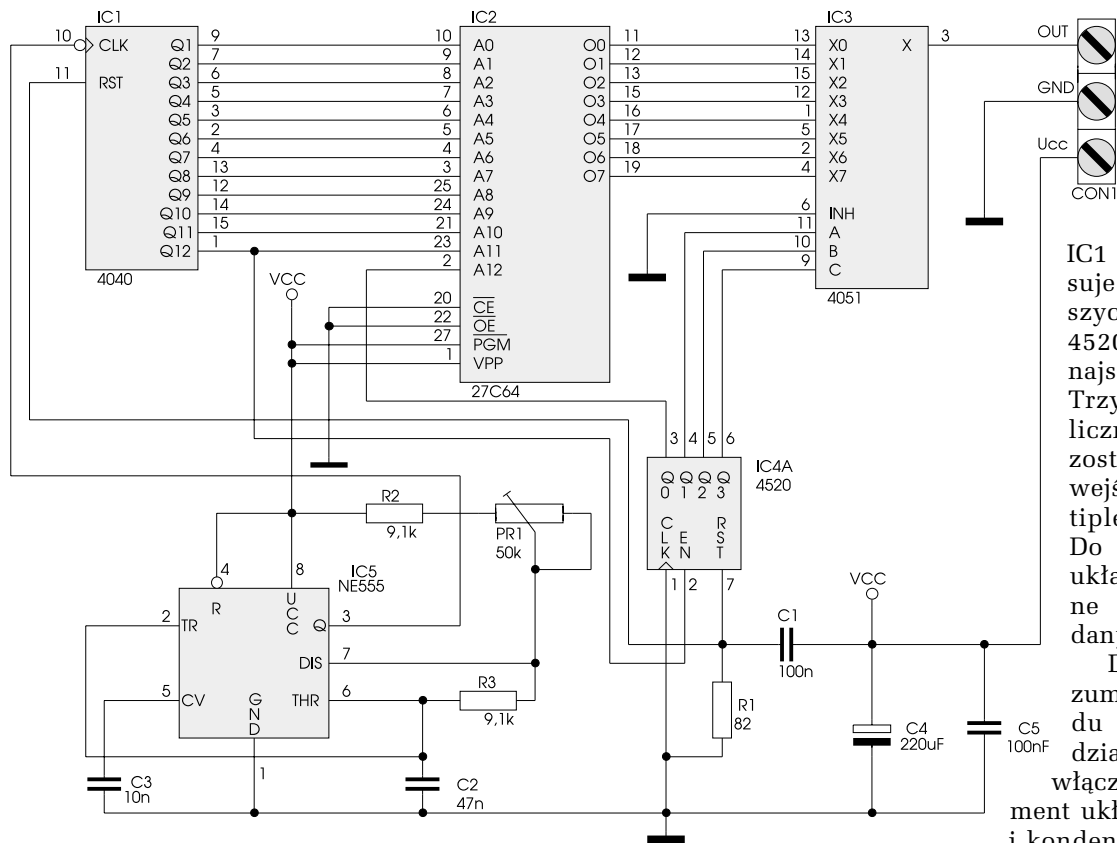
Sygnał DCF77 może być także wykorzystywany do korygowania zegara systemowego w komputerach klasy PC. I w tym przypadku napisanie odpowiedniego programu nie jest czynnością trudną. Mimo że nie uważam się za dobrego programistę, to taki program udało mi się napisać w BASIC-u w ciągu kilku godzin.

Nie ma jednak róży bez kolców. Takim „kolcem” przy wykorzystywaniu sygnału DCF77 są zmienne warunki propagacji sygnału radiowego i wynikające z tego zjawiska okresowe zaniki transmisji sygnału DCF77. Podczas normalnej pracy zegara nie ma to zbyt wielkiego znaczenia, ponieważ zupełnie wystarczające jest korygowanie czasu jedynie co pe-

wien czas, np. raz na kilka godzin. Także zegar systemowy w komputerze, pomimo jego zwykle marnych parametrów, wymaga w praktyce korekcji raz na dobę. Sytuacja komplikuje się jednak w przypadku, kiedy testujemy wykonany układ zegara lub napisany program. Jeżeli zegar nie „załapuje” właściwego czasu, to przyczyny mogą być dwie: błąd konstruktora lub programisty albo nieodebranie poprawnego kodu transmisji.

Stwierdzenie, która z tych przyczyn powoduje niewłaściwą pracę układu jest zwykle dość trudne. Wprawdzie większość odbiorników DCF77 posiada sygnalizację optyczną odbierania sygnału, ale stwierdzenie poprawności transmisji „na oko” jest praktycznie niemożliwe. Ponieważ dość często zajmuję się uruchamianiem i testowaniem zegarów DCF77, zdecydowałem się na zbudowanie niezwykle prostego i taniego układu, który radykalnie może usunąć opisane wyżej problemy i znacznie ułatwić życie konstruktorowi lub programiście pragnących wykorzystywać możliwości jakie daje atomowy wzorzec czasu z Niemiec.

Budowanie urządzenia, którego jedynym zadaniem byłoby testowanie zegarów DCF77 nie miałoby jednak w wielu przypadkach ekonomicznego uzasadnienia. Jednak proponowany układ będzie można wykorzystać także do innych celów, o czym wspomnimy jeszcze w dalszej części artykułu.



Rys. 1. Schemat elektryczny urządzenia.

Opis działania układu

Na rys. 1 został pokazany schemat elektryczny proponowanego układu. Zanim jednak przejdziemy do jego analizy, wspomniemy w największym skrócie o transmisji DCF77 i stosowanym w niej kodzie. Szczegółowy opis DCF77 został zamieszczony w numerze EP3/97, w artykule opisywanym „Mówiący zegar z DCF77”.

Sygnal DCF77 jest nadawany z terenu Niemiec na częstotliwości „ultradługiej” 77,5 kHz i swoim zasięgiem obejmuje prawie całą Europę, w tym całe terytorium Polski.

Ramka danych w sygnale DCF77 zawiera następujące informacje:

- aktualny czas w godzinach, minutach i sekundach;
- aktualną datę: rok, miesiąc, dzień miesiąca i dzień tygodnia;
- informację o aktualnym czasie: zimowym lub letnim.

Informacja jest przekazywana pod postacią kodu zero-jedynkowego o długości 60 bitów i czasie trwania 60 s. Kod jest przekazywany w logice dodatniej, impuls o czasie trwania 100 ms oznacza „0”, a impuls o czasie trwania 200 m - „1”. Znaczenie poszczegól-

nych bitów zostało omówione w tab. 1.

Dane zawarte w powyższej tabeli pozwolą Czytelnikom łatwiej zrozumieć zasadę działania proponowanego układu i ewentualnie własnoręcznie zaprogramować EPROM potrzebny do jego działania. Po pozostałe informacje na temat DCF77 odsyłamy Czytelników do lektury wspomnianego wyżej artykułu.

Łatwo zauważyć, że sercem układu jest pamięć EPROM typu 2764 o pojemności 64 kilobitów. Tak, właśnie kilobitów, ponieważ tym razem pojemność tej pamięci musimy wyrazić w tych jednostkach, a nie jak zwykle w bajtach.

Zastosowanie takiej pamięci może wydać się pozbawione sensu: przecież pamięć taka przechowuje informację w postaci słów ośmiobitowych, a nam potrzebny jest tylko ciąg impulsów zero-jedynkowych. Odpowiedniejsze byłoby zastosowanie pamięci operującej słowem jedynbitowym, ale pamięci takie są dość trudne do zdobycia i stosunkowo drogie, znacznie droższe od powszechnie stosowanych pamięci z rodziny 27XXX. Za chwilę okaże się, w jak prosty sposób „zmusiliśmy”

2764 do zapisywania informacji w słowach jedynbitowych.

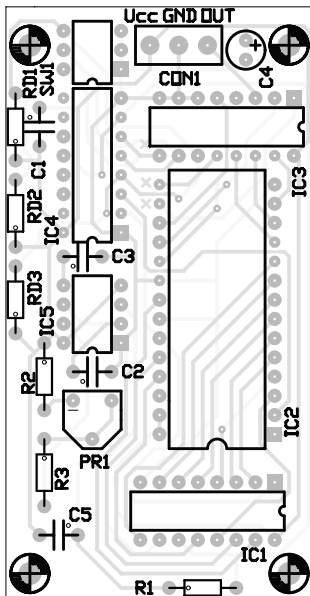
Pamięć IC2 jest adresowana za pomocą dwóch liczników:

IC1 - 4040, który adresuje dwanaście młodszych bitów i IC4A - 4520, który „obsługuje” najstarszy bit adresu. Trzy pozostałe wyjścia licznika binarnego IC4A zostały dołączone do wejść adresowych multiplexera IC3 - 4051. Do wejść danych tego układu zostały dołączone wszystkie wyjścia danych pamięci IC2.

Dla łatwiejszego zrozumienia budowy układu prześledźmy jego działanie od momentu włączenia zasilania. Fragment układu z rezystorem R1 i kondensatorem C1 służy wyzerowaniu obydwu liczn-

ków w momencie rozpoczęcia przez układ pracy (włączenia zasilania). Generator astabilny zbudowany z wykorzystaniem mojej ulubionej kostki - NE555 (IC5), wytwarza ciąg impulsów prostokątnych o częstotliwości regulowanej za pomocą potencjometru montażowego PR1. W podstawowym zastosowaniu układu częstotliwość ta powinna wynosić 1kHz.

Po wyzerowaniu liczniki rozpoczynają pracę, adresując kolejne bajty pamięci IC2. Warto zauważyć, że w ciągu pierwszego cyklu zliczania, na wejściach adresowych multiplexera IC3 panuje stan logiczny „000” i na wyjście tego układu przekazywana jest informacja z wejścia X0, czyli z najmłodszego wyjścia danych pamięci IC2. Pierwszy cykl zliczania kończy się w momencie osiągnięcia przez licznik IC1 stanu „11111111110”, a przez licznik IC4A stanu „0010”. Cała zawartość pamięci IC2 została odczytana, ale na wyjście OUT układu przekazane zostały jedynie dane z jej najmłodszego wyjścia danych (najmniej znaczący bit każdego słowa). W tym momencie odczytywanie zawartości pamięci rozpocznie się od początku, z tym, że na wyjście



Rys. 2. Rozmieszczenie elementów na płytce drukowanej.

układu będzie teraz przekazywana informacja z wyjścia Q1 IC2.

Cały opisany wyżej cykl będzie powtarzał się ośmiokrotnie i na wyjście układu zostaną przekazane wszystkie 64 kilobity zawartości pamięci, w której zapisano kod DCF77. Sądzę, że do pełniejszego zrozumienia zasady działania urządzenia może przyczynić się opis programowania pamięci zamieszczony w dalszej części artykułu.

Montaż i uruchomienie

Na rys. 2 zostało przedstawione rozmieszczenie elementów na płytce drukowanej. Ze względu na znaczne skomplikowanie połączeń, płytka została wykonana na laminacie dwustronnym z metalizacją otworów.

Uważni Czytelnicy z pewnością zauważyli już pewne rozbieżności pomiędzy schematem a rysunkiem płytki, na której widać jakieś nieznanne elementy. Cierpliwości, zaraz sobie wszystko wyjaśnimy.

Montaż układu wykonujemy w typowy sposób, rozpoczynając od elementów o najmniejszych gabarytach, a kończąc na największych. Pod układy scalone warto zastosować podstawki (pod układ pamięci IC2 podstawa jest właściwie niezbędna). Układ zmontowany ze sprawdzonych elementów nie wymaga oczywiście uruchamiania, ale jedynie regulacji częstotliwości pracy generatora zegarowego, która powinna wynosić możliwie dokładnie 1kHz. Układ wymaga zasilania napięciem

5VDC, typowym dla układów TTL, koniecznym stabilizowanym.

Na rys. 3 przedstawiono sposób dołączenia wykonanego układu do „Mówiącego zegara z DCF77” - AVT322. Dołączenie układu do innych zegarów powinno być zgodne z ich specyfikacją techniczną.

W prezentowanym kicie będzie dostarczana zaprogramowana pamięć EPROM. Po podłączeniu układu do sprawnego zegara i odczytaniu kodu, na wyświetlaczach pokaże się godzina i data 14:24, 14-12. O tym, jakie to będą wartości powiemy za chwilę. Na razie zajmijmy się innym problemem. Z pewnością wielu Czytelników zechce samemu zaprogramować sobie EPROM i należy się im kilka wskazówek, jak mają to uczynić.

Najpierw musimy ustalić, jakie dane muszą zostać zapisane w EPROM-ie. W tab. 2 przedstawiono przykładowy ciąg impulsów, jaki musi być zapisany w pamięci, który po przekazaniu do zegara DCF77 spowoduje wyświetlenie czasu i daty. Tu Czytelnicy są proszeni o samodzielne przeanalizowanie tabeli i obliczenie jaka godzina i data zostanie wyświetlona na zegarze po przekazaniu do niego tego kodu.

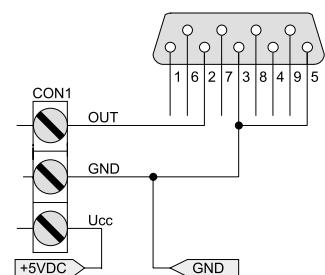
A więc wiemy już, jaki ciąg impulsów musimy zarejestrować w EPROM-ie. Wiemy także, że mamy do dyspozycji dokładnie 65536 bitów, a częstotliwość zegarowa wynosi 1000Hz. A zatem na każdy impuls przypadnie 1000 bitów (część pamięci pozostanie niewykorzystana i zapisana samymi zerami) co daje bardzo dużą częstotliwość próbkowania i zapewnia dużą dokładność przekazywanej informacji.

Obliczenie potrzebnych wartości „na piechotę” byłoby niezwykle trudne i pozbawione większego sensu. Przecież aby zaprogramować pamięć musimy mieć do dyspozycji komputer, a zatem i potrzebne obliczenia możemy wykonać za pomocą arkusza kalkulacyjnego. Do przygotowania programu używałem arkusza MS EXCEL, ale można zastosować dowolny inny arkusz, np. LOTUS 1-2-3. Pierwszym krokiem będzie ponumerowanie komórek arkusza, co bardzo ułatwi nam wpisywanie danych. Przy wykorzystywaniu zautomatyzowanej funkcji „Wypełnij serią danych” ta czynność

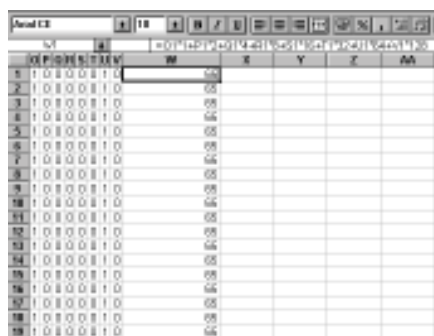
Tabela 1.

Kolejny impuls	Znaczenie
0..14	Początek transmisji, wszystkie "0"
15	Typ anteny
16	Zmiana czasu, na godzinę przed zmianą "1"
17..18	Typ czasu - "01" czas zimowy, - "10" - czas letni
19	Korekta czasu, "1" zapowiedź korekty
20	Początek informacji o aktualnym czasie, zawsze "1"
21..24	Jednostki minut w kodzie BCD
25..27	Jednostki godzin w kodzie BCD
28	Bit parzystości, "0" przy parzystej liczbie "jedynek" w kodzie minut
29..32	Jednostki godzin w kodzie BCD
33, 34	Dziesiątki godzin w kodzie BCD
35	Bit parzystości, "0" przy parzystej liczbie "jedynek" w kodzie godzin
36..39	Jednostki dnia miesiąca w kodzie BCD
40, 41	Dziesiątki dnia miesiąca w kodzie BCD
42..44	Dzień tygodnia w kodzie BCD, 1 - Poniedziałek
45..48	Jednostki miesiąca w kodzie BCD
49	Dziesiątki miesiąca w kodzie BCD
50..53	Jednostki roku w kodzie BCD
54..57	Dziesiątki roku w kodzie BCD
58	Bit parzystości, Bit parzystości, "0" przy parzystej liczbie "jedynek" w kodzie daty
59	Bez impulsu

nie zajmie nam więcej niż minutę. Kolejnym krokiem będzie wpisanie na początku każdego tysiąca bitów serii jedynek: "sto" dla logicznego zera i "dwieście" dla logicznej jedynki. Następnie trzeba będzie ułożyć zapisane komórki w osiem kolumn, każda po 8192 wierszy i umieszczenie w sąsiedniej kolumnie formuły przeliczającej wartość komórek w każdym z wierszy na postać binarną. Fragment arkusza kalkulacyjnego z wpisaną formułą i danymi widoczny jest na rys. 4. Ostatnią czynnością będzie zapisanie w arkuszu wyników obliczeń metodą „Kopiuj, Wklej specjalnie, jako Wartości” przeniesienie ich do pli-



Rys. 3. Sposób dołączenia emulatora do zegara AVT-322.



Rys. 4. Okno Excela z opisem zawartości pamięci EPROM.

ku tekstowego i przekształcenie w postać binarną czytelną dla programatorów EPROM.

Korzystanie z wykonanego urządzenia jest niezwykle proste. Za pomocą właściwego dla testowanego urządzenia złącza podłączamy nasz układ do zegara lub komputera. W minutę po włączeniu zasilania układu na wyświetlaczach zegara powinna pojawić się godzina i minuta. W układzie modelowym i przy korzystaniu z EPROM-a dostarczanego w kicie będzie to godzina 14:24. Po przełączeniu zegara w tryb wyświetlania daty na wyświetlaczach wyświetlona zostanie data: 14-12. Oczywiście, wartości czasu i daty zaprogramowane w EPROM-ie są całkowicie

Tab. 2.

Impuls	Wartość	Impuls	Wartość
0	0	31	1
1	0	32	0
2	0	33	1
3	0	34	0
4	0	35	0
5	0	36	0
6	0	37	0
7	0	38	1
8	0	39	0
9	0	40	1
10	0	41	0
11	0	42	0
12	0	43	1
13	0	44	0
14	0	45	0
15	0	46	0
16	0	47	1
17	0	48	0
18	1	49	1
19	0	50	1
20	1	51	0
21	0	52	0
22	0	53	0
23	1	54	1
24	0	55	0
25	0	56	0
26	1	57	0
27	0	58	1
28	0	59	0
29	0	60	0
30	0	61	0

dowolne. Wystarczy, że znamy ich wartość i dzięki temu możemy sprawdzić poprawność przetwarzania danych przez układ zegara lub napisany program komputerowy. Jeżeli nasz układ pozostanie dołączony do testowanego urządzenia przez dłuższy okres, to zajdzie bardzo śmieszne zjawisko: wyświetlany czas może zmienić się najwyżej o minutę i to na kilka sekund. Po tym czasie powtórnie odebranie transmisji DCF77 z naszego układu „cofnie“ zegar do początkowej wartości czasu (oczywiście, stanie się tak tylko w wypadku zegara z permanentną korekcją aktualnego czasu, np. z zegarem AVT-322).

Interesującą wydaje się być jeszcze jedna możliwość zastosowania wykonanego układu. Konstruktorzy mają trudności nie tylko ze sprawdzeniem poprawności działania wykonanego zegara DCF77, ale i z testowaniem odbiornika służącego odbieraniu transmisji z Mainflingen. Tu także nie wiadomo czy odbiornik działa wadliwie, czy też złe warunki propagacji fal radiowych uniemożliwiają mu odebranie kodu transmisji. Wystarczy tylko dobudować do naszego układu prosty nadajniczek radiowy o zasięgu maksymalnie kilku metrów i sprawa załatwiona!

Pojawia się jednak jeden problem: od początku mojej kariery konstruktora miałem wrodzony wstręt do wszelkiego rodzaju cewek, a sama wzmianka o strojonej indukcyjności może mnie przyprowadzić o kolejny atak serca. Nigdy w życiu nie potrafiłem zbudować nawet najprostszego odbiornika radiowego, nie mówiąc o nadajnikach. Ale może ktoś z Szanownych Czytelników potrafi wykonać taki nadajnik i rezultatami swojej pracy zechce podzielić się z nami na łamach działu „Projekty Czytelników“?

Pozostała nam jeszcze ostatnia sprawa do załatwienia: wyjaśnienie roli, jaką pełnią dodatkowe elementy na płytce obwodu drukowanego. Jak już wspominałem, opisany układ może służyć także do innych celów niż testowanie zegarów DCF. Może zostać zastosowany wszędzie tam, gdzie potrzebne jest zarejestrowanie i wielokrotne odtwarzanie długich ciągów impulsów zero-jedynkowych, np. do nadawanie

WYKAZ ELEMENTÓW

Rezystory

- PR1: 50kΩ
- R1: 82kΩ
- R2, R3: 9,1kΩ

Kondensatory

- C1, C5: 100nF
- C3: 10nF
- C2: 47nF
- C4: 220µF/16V

Półprzewodniki

- IC1: 4040
- IC2: 27C64 zaprogramowana pamięć EPROM
- IC3: 4051
- IC4: 4520
- IC5: NE555

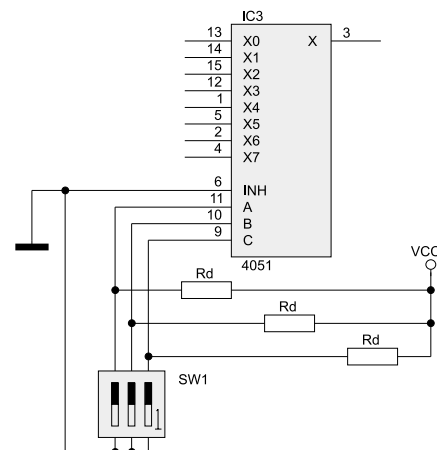
Różne

- CON1: mini ARK3 (3,5mm)

stałych komunikatów za pomocą alfabetu Morse'a.

Nie zawsze jednak tak długi ciąg znaków będzie potrzebny i dlatego postanowiłem zapewnić Wam możliwość podzielenia obszaru pamięci na osiem niezależnych banków i wybierania ich za pomocą przełącznika. Jest to właśnie przełącznik oznaczony na płytce jako SW1 i nie uwidoczniiony na schemacie głównym. Na rys. 5 został pokazany fragment układu, który może być zmodyfikowany. W takiej wersji, połączenia pomiędzy wyjściami: Q1..Q3 - IC4A i wejściami: A, B i C - IC3 muszą być przerwane (przecięcie ścieżek w miejscach oznaczonych na płytce symbolem „x“). Rezystory oznaczone jako Rd służą do „podciągnięcia“ do plusa zasilania wejść nie zwartych z masą przez przełącznik SW1, a ich wartość nie jest krytyczna (1..30kΩ).

Zbigniew Raabe, AVT



Rys. 5. Sposób podzielenia pamięci EPROM na partycje.