

Analogowo-cyfrowy analizator widma

Analizator, jak wskazuje sama nazwa, służy do analizowania i wyświetlania widma sygnału audio. Projekt analizatora widma był już przedstawiany (w EdW 04/2001). Był to ciekawy projekt, ale o wiele bardziej skomplikowany (sprzętowo), mniej poddawał się modyfikacjom oraz wymagał symetrycznego zasilania +/-12V.

Przedstawiony projekt nie jest zwykłym remakiem analizatora analogowego. Zrealizowanie go wymaga sporej wiedzy teoretycznej na temat teorii sygnałów, a także pewnej wiedzy praktycznej, aby umieć poradzić sobie ze specyficznymi problemami, niespotykanymi w układach analogowych.

Chyba dlatego też do tej pory nikt na łamach EdW nie dotykał tej dziedziny. Przedstawiony projekt jest jednym z pierwszych układów realizujących cyfrową obróbkę sygnału, czyli DSP!!! (DSP – Digital Signal Processing).

Ze względu na duży stopień skomplikowania programu projekt oznaczony jest 3 gwiazdkami. Jednak wgłębianie się w zawoilości programu nie jest niezbędne do zrozumienia zasady działania ani tym bardziej do zbudowania i uruchomienia analizatora, dlatego także początkujący będą w stanie zbudować i przetestować to urządzenie.

Opis układu

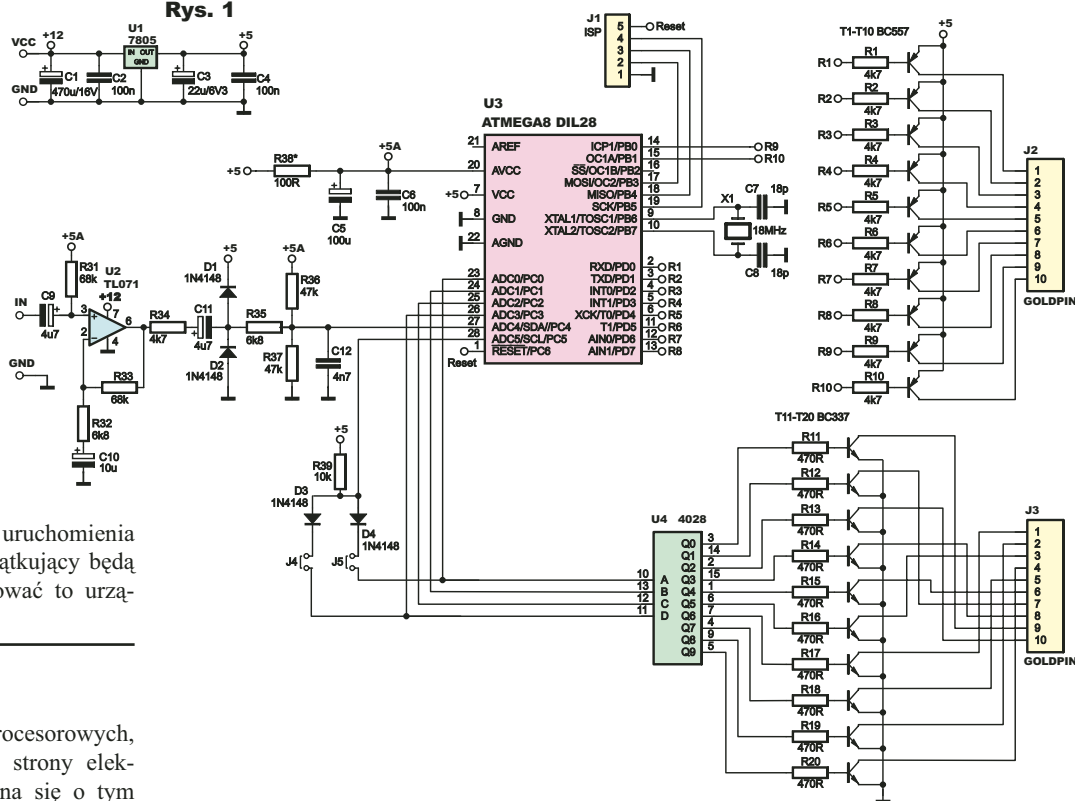
Część sprzętowa

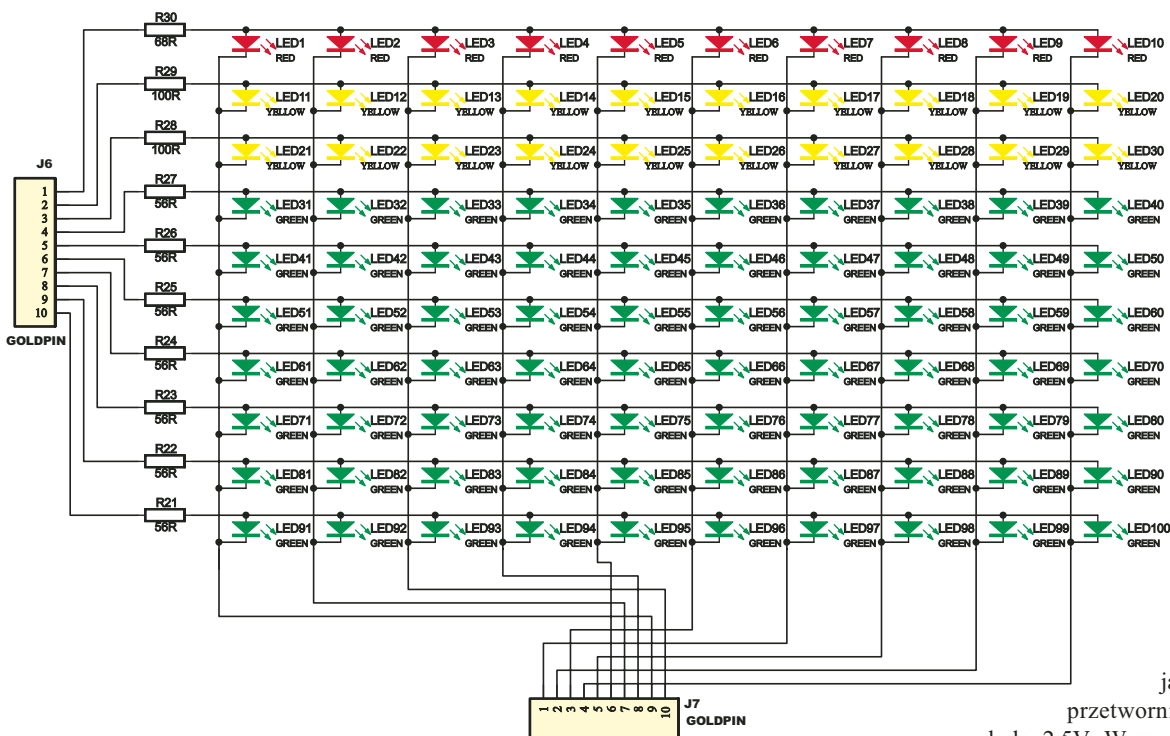
Jak większość układów mikroprocesorowych, przedstawiony projekt jest od strony elektronicznej bardzo prosty. Można się o tym

przekonać, patrząc na schemat płyty głównej przedstawiony na **rysunku 1** i schemat wyświetlacza – na **rysunku 2**. Napięcie zasilające o wartości 9...16V jest podawane na punkty VCC i GND. Jest ono filtrowane przez kondensatory C1 i C2 i trafia do stabilizatora U1. Napięcie z wyjścia stabilizatora, blokowane kondensatorami C3 i C4, zasila przede wszystkim mikrokontroler U3. Ze względu na uproszczenie układu, wyświetlacz złożony z diod LED także jest zasilany stabilizowanym napięciem 5V. Gdybyśmy chcieli zasilić je innym napięciem, nieważne czy stabilizo-

wanym, czy nie, musielibyśmy zastosować transformatory poziomów napięć, aby przesłać co najmniej kilka sygnałów cyfrowych do wyświetlacza. Zastosowanie wspólnego napięcia zasilania rozwiązuje ten problem, ale niestety stwarza inne: mianowicie stabilizator obciążony dużym prądem wydziela duże ilości ciepła i wymaga zastosowania radiatora. Jednak wydzielana na nim moc nie przekracza kilku watów i już mała blaszka o powierzchni 5...10cm² jest w stanie zapewnić wystarczające chłodzenie. Bez radiatora albo ze zbyt małym radiatorem analizator od czasu

Rys. 1





Rys. 2

do czasu będzie się wyłączał. Drugim problemem jest fakt, że diody są zasilane prądem impulsowym, a to powoduje powstawanie dużych zakłóceń w szynie zasilającej część analogową (głównie odczuwa to przetwornik analogowo-cyfrowy zawarty we wnętrzu mikrokontrolera). Zastosowanie prostego filtra w postaci rezystora R38 i kondensatora C5 rozwiązały ten problem, gdyby nie pewna wada mikrokontrolera ATmega8, polegająca na tym, że projektanci tej kostki pomyłkowo połączyli wewnątrz struktury szyny VCC i AVCC! Rezystancja połączenia to około 50Ω, co łatwo sprawdzić multimetrem. Co ciekawe, wada ta do tej pory nie została usunięta, więc nawet ostatnio wyprodukowane procesory tego typu ją zawierają. I nawet najnowsza nota katalogowa nie zawiera żadnej wzmianki o niej! Wada ta na szczęście dotyczy jedynie kostki ATmega8. Nie dotyczy już jej nowszej wersji, czyli ATmega88 ani żadnego innego procesora AVR. Jedynym sposobem pozbycia się zakłóceń na szynie zasilającej część analogową jest zastosowanie kondensatora filtrującego o dużej pojemności (albo zastosowanie procesora ATmega88). Rolę tę pełni C5 o pojemności aż 100μF. Pojemność 100μF naprawdę jest duża, bo przetwornik ADC i kilka innych elementów podłączonych do analogowej szyny zasilania pobierają znacznie mniej niż 1mA prądu. Rezystor R38 oznaczony gwiazdką nic nie wnosi do układu, bo szyny VCC i AVCC i tak są już połączone. Rezystor ten nie będzie więc montowany.

Sygnal audio trafia na punkt IN i za pośrednictwem kondensatora C9 przedostaje się na wejście nieodwracającej wzmacniacza operacyjnego U2. Do zapewnienia przyzwo-

itej dokładności analizatora wymagane jest zapewnienie możliwie dużego wysterowania wejścia przetwornika ADC, jednak bez przesterowania. Wymagałoby to zastosowania wzmacniacza z wyjściem typu Rail-to-Rail, ale wzmacniacze takie nie są zbyt popularne.

Wykorzystany wzmacniacz operacyjny typu TL071 jest w stanie wytworzyć na wyjściu sygnał o wartości międzyszczytowej równej 5V, a nawet i większej, ale pod warunkiem że będzie zasilany napięciem dużo wyższym niż 5V. Praca z wyższym napięciem zasilania, szczególnie niestabilizowanym, stwarza jednak ryzyko przesterowania, dlatego konieczne jest zastosowanie ogranicznika na wyjściu wzmacniacza. Przesterowanie, a w zasadzie „wystawianie” sygnału poza zakres napięcia zasilającego 5V jest niebezpieczne, bo może doprowadzić do uszkodzenia wejścia przetwornika ADC. W nocie katalogowej mikrokontrolera znajduje się informacja, że napięcie na żadnym pinie nie powinno nigdy być niższe od napięcia masy bardziej niż o 0,5V ani wyższe bardziej niż o VCC+0,5V. Samo przesterowanie rozumiane jako obcinanie wierzchołków sygnału nie jest już niebezpieczne, ale powoduje powstawanie znacznych ilości harmonicznych, co oczywiście także jest niepożądane.

W największym skrócie wzmacniacze serii TL07x mogą wytworzyć na swym wyjściu minimalne napięcie równe około 2V, licząc od strony ujemnego napięcia zasilania, a maksymalnie o około 2V niższe od górnego napięcia zasilania. Przetwornik ADC wymaga napięcia o wartości międzyszczytowej równego (maksymalnie) 5V, czyli takiego o amplitudzie 2,5V. Masa wzmacnianego sygnału znajdu-

je się na poziomie 5V, więc z prostego rachunku wynika, że aby wzmacniacz był w stanie wytworzyć sygnał o amplitudzie 5V, musi być zasilany napięciem co najmniej 9,5V. Tak więc układ TL071 wymaga większego napięcia zasilania niż mikrokontroler, dlatego też jest zasilany napięciem sprzed stabilizatora U1.

Ogranicznik składa się z rezystorów R34, R35, R36, R37 i diod D1, D2. Rezystory R36 i R37 tworzą dzielnik napięcia polaryzujący wstępnie wejście przetwornika ADC napięciem

około 2,5V. Wzmocniony sygnał audio z wyjścia wzmacniacza operacyjnego U2 jest za pośrednictwem kondensatora C11 podawany na ten dzielnik i moduluje napięcie na nim. Gdy pojawi się za duży sygnał, zostanie on zwarty przez diody D1 i D2. Rezystor R34 chroni przed zwarcieniem w momencie, gdy sygnał jest ograniczany. Jednak taki prosty diodowy ogranicznik zapewnia ograniczenie sygnału do wartości $-0.7V \dots VCC+0.7V$. Teraz ujawnia się rola rezystora R35. Wraz z rezystorami R36 i R37 tworzy on dzielnik napięcia, który nieznacznie tłumi sygnał. Wartości rezystorów zostały tak dobrane, że napięcie na wejściu przetwornika ADC zawiera się w przedziale prawie równym $0 \dots 5V$. Wprawdzie mikrokontrolery ATmega8 są wyjątkowo odporne na uszkodzenia, o czym się wielokrotnie przekonałem, ale praca z często pojawiającym się zbyt niskim lub zbyt wysokim napięciem na wejściu na pewno jest ryzykowna, i dlatego jest tutaj trochę bardziej rozbudowany ogranicznik.

Kondensator C12 wraz z szeregowo połączonymi rezystorami R34 i R35 tworzy prosty filtr antyaliasingowy.

Niestety kostka ATmega8 ma za małą liczbę nóżek, aby bezpośrednio obsłużyć wyświetlacz składający się z 10 kolumn i 10 wierszy, odebrać dane analogowe audio i odczytać stan jumperów J4 i J5. Dałoby się to jeszcze jakoś zrealizować, gdyby nie konieczność zajęcia 2 pinów przez rezonator X1. Dlatego też zastosowano dekodery 4028, który zwiększa liczbę pinów wyjściowych. Transzystory T1...T10 i T11...T20 zwiększają wydajność prądową tych pinów.

Diody D3 i D4 zabezpieczają przed zwarcieniem w momencie, gdy włożone są oba jumperzy J4 i J5. Rezystor R39 ze względu na



dużą szybkość odświeżania wyświetlacza jest konieczny, aby zapewnić odpowiednio szybkie i stabilne stany logiczne na pinie PORTC.5.

Część programowa

Jak widać, część sprzętowa jest rzeczywiście bardzo prosta, co niestety nie zapewnia prostoty części programowej. Część analogowa jedynie odpowiednio wzmacnia sygnał i zabezpiecza przed przesterowaniem, ale w zasadzie nic poza tym nie wnosi, więc na wejściu przetwornika ADC dostępny jest nieobrobiony sygnał audio. Cały proces wyznaczania widma sygnału jest przerzucony na procesor. Każdy, kto kiedykolwiek próbował napisać jakikolwiek program przetwarzający sygnały cyfrowe, już na pewno się domyśla, jak bardzo skomplikowany będzie program sterujący procesorem. Program ze względu na wymaganą dużą szybkość wykonywania jest napisany w assemblerze, więc niestety nie będzie łatwy do zrozumienia. Żeby przynajmniej wyjaśnić zasadę filtrowania zastosowaną w przedstawionym analizatorze, wyjaśnię przynajmniej, jak działają zastosowane tutaj filtry IIR.

Filtry IIR (Infinite Impulse Response) są mocno związane z filtrami analogowymi. W zasadzie filtry IIR są matematyczną symulacją działania analogowych filtrów aktywnych. Istnieją filtry dolnoprzepustowe, środkowo-przepustowe i górnoprzepustowe. Można dowolnie ustalać ich częstotliwość graniczną, dobroć i wzmocnienie. Skoro są zrealizowane cyfrowo, to z oczywistych względów nie zmieniają swoich parametrów wraz z temperaturą. Dzięki temu nie będą się wzbudzały przy zmianach temperatury itp. Są chyba naj-

prostszymi w działaniu filtrami cyfrowymi. Dzięki temu nie narzucają dużych wymagań na procesor, na którym mają pracować. Mają oczywiście także i wady. Nie są w stanie skutecznie odfiltrować przebiegów o częstotliwościach większych od połowy częstotliwości próbkowania. Nie jest to już wada samych filtrów, tylko niekorzystne zjawisko aliasingu. Przy danej dokładności obliczeń istnieje określona minimalna częstotliwość, na którą zaprojektować poprawnie działający filtr. Dość trudne do zaprojektowania są filtry o dużych częstotliwościach (dużych, czyli takich, które zbliżają się do połowy częstotliwości próbkowania).

Schematycznie filtr IIR działa tak jak to przedstawiono na rysunku 3. Do obliczenia kolejnej próbki wyjściowej jest potrzebna „przedpoprzednia” próbka wejściowa oraz dwie wcześniejsze próbki wyjściowe. Przykładowy kod obliczający kolejne próbki jest przedstawiony na listingu 1. Współczynniki α , β i γ są obliczane z dość skomplikowanych wzorów. Można je też wyznaczać doświadczalnie, jeżeli posiadamy program pozwalający wyznaczać charakterystykę filtru dla zadanych parametrów α , β i γ . Ja sam, mając problemy z wyznaczeniem współczynników dla kanałów 4kHz i 8kHz, wyznaczyłem je doświadczalnie, bowiem charakterystyki filtrów „zbudowanych” w oparciu o obliczone współczynniki bardzo różniły się od zamierzonych. Filtr na częstotliwość 16kHz ze względu na wymaganą bardzo dużą szybkość działania zrealizowałem dużo prościej. Po prostu każda próbka wyjściowa tego filtru jest różnicą dwóch sąsiednich próbek wejściowych. Taki prosty filtr to filtr górnoprzepustowy. Mikrokontroler mimo prostoty filtrów nie jest w stanie rów-

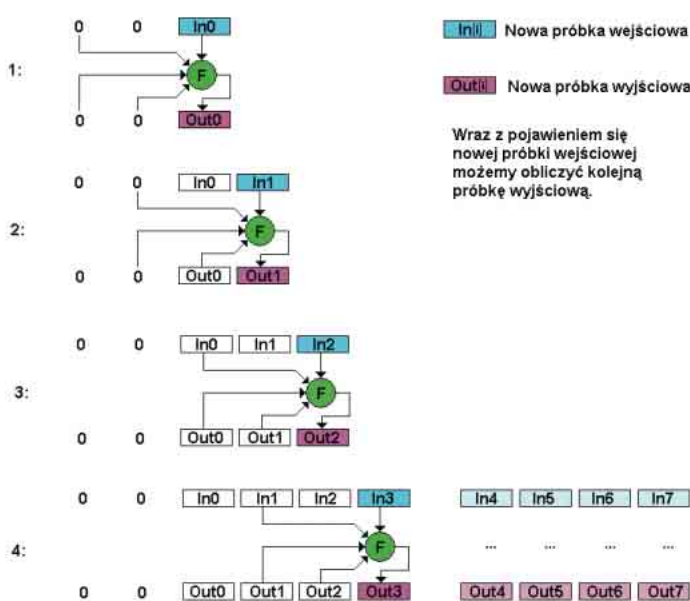
nocześnie zrealizować aż 10 filtrów. Dlatego zastosowałem pewną sztuczkę. Polega ona na tym, że zmniejszamy częstotliwość próbkowania dla filtrów o niższych częstotliwościach. Jest to realizowane przez uśrednianie wartości dwóch próbek w celu uzyskania jednej próbki (rysunek 4). Dzięki temu uzyskujemy kilka

```
Dim Buffer_In(100) as Single
Dim Buffer_Out(100) as Single
Dim I as Byte
Dim J as Byte
Dim K as Byte
Dim Zmienna1 as Single
Dim Zmienna2 as Single

Const Alfa = 0.0461841095
Const Beta = 0.407631781
Const Gamma = 0.871031797

Buffer_In(1) = 0 'Wymagane jest
Buffer_In(2) = 0 'takie
zainicjowanie
Buffer_Out(1) = 0 'tych zmiennych
na
Buffer_Out(1) = 0 'początku
obliczeń
..... 'Tutaj trzeba wpro-
wadzić próbki 'danych do przefil-
trowania. Pierwsza 'wprowadzana
próbka ma być w zmiennej 'Buffer_
In(3)! Robimy tak dlatego, bo
'pierwsze dwie próbki (Buffer_
In(1) i 'Buffer_In(2) mają mieć
wartość 0.
'Obliczenia prowadzimy zaczynając
od 'próbki drugiej (czyli od
Buffer_In(3))
For I=3 to 100
    J=I - 1
    K=I - 2
    Zmienna1=Buffer(I) - Buffer(K)
    Zmienna2 = Zmienna1 * Alfa
```

Rys. 3

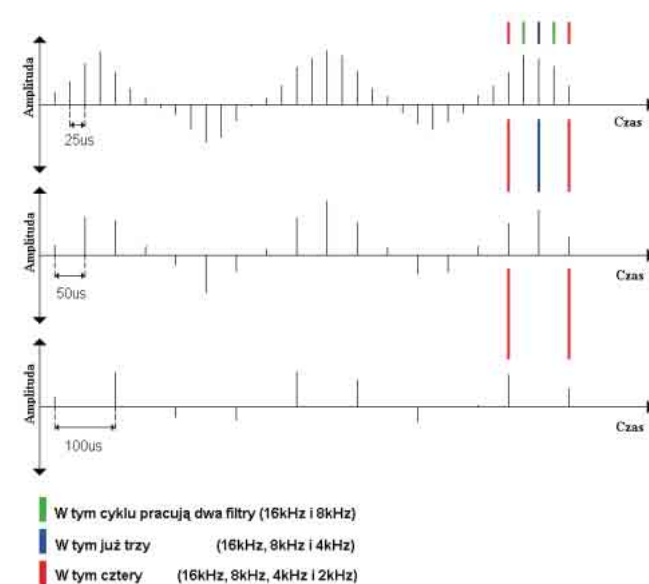


Funkcja pozwalająca policzyć kolejną próbkę wyjściową:

$$F: \text{Out}[x] = \alpha \cdot \text{In}[x] - \text{In}[x-2] + \gamma \cdot \text{Out}[x-1] - \beta \cdot \text{Out}[x-2]$$

Listing 1

Rys. 4



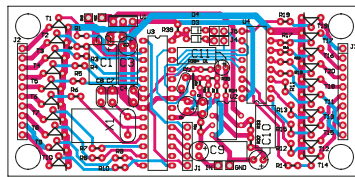
Oczywiście istnieją i takie cykle w których pracuje jeszcze więcej filtrów, na przykład wszystkie 10. Takie cykle najbardziej obciążają procesor obliczeniami. Na szczęście występują one jeszcze rzadziej niż te przedstawione na rysunku.

strumieni o coraz mniejszych częstotliwościach próbkowania. I co za tym idzie, każdy filtr o coraz mniejszej częstotliwości jest uruchamiany coraz rzadziej. I tak dla przykładu filtr pracujący na 16kHz przy częstotliwości próbkowania 40178Hz jest uruchamiany 40178 razy w ciągu sekundy. Tak samo filtr na częstotliwość 8kHz, bo on też pracuje przy częstotliwości próbkowania 40178kHz. Wszystkie filtry począwszy od filtru 4kHz aż do 31Hz pracują już na wolniejszych strumieniach. Filtr 4kHz filtruje strumień o częstotliwości próbkowania 20089Hz, czyli jest uruchamiany tylko 20089 razy na sekundę, dwa razy rzadziej niż filtry 8 i 16kHz; filtr 2kHz około 10044 razy na sekundę, a na przykład filtr na 31Hz tylko 314 razy na sekundę. Daje to przyspieszenie filtrowania o około 1/3. Taki proces filtrowania można porównać do filtrowania dźwięku zapisanego na taśmie, lecz odzwierciedlanego w zwolnionym tempie.

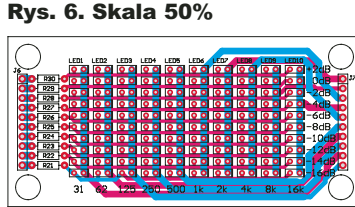
Niestety taka sztuczka powoduje znowu pojawienie się zjawiska aliasingu w wolniejszych strumieniach, tym razem aliasingu trudnego do wyeliminowania. Dlatego analizator pracujący ze zbyt dużym sygnałem audio (czyli z przesterowaniem) zaczyna ujawniać swoje wady. Filtry o niższych częstotliwościach zaczynają delikatnie reagować na kilkakrotnie wyższe częstotliwości. Na szczęście wada ta ujawnia się dopiero przy mocnym przesterowaniu. Jednak jeżeli ktoś myśli o profesjonalnym zastosowaniu analizatora, to powinien pamiętać o tej wadzie.

Współczynniki większości filtrów są identyczne, bo wraz z częstotliwością filtru zmniejsza się także częstotliwość próbkowania. Na przykład filtr 62Hz pracuje przy częstotliwości próbkowania 628Hz, a filtr 31Hz na 314Hz. Czyli w obu przypadkach stosunek częstotliwość filtru/częstotliwość próbkowania jest taki sam oraz dobroć i wzmocnienie filtrów ma być takie samo, więc i wszystkie współczynniki α , β i γ są takie same.

W przerwaniu od Timera2 jest umieszczony kod odpowiadający za uruchamianie pomiaru za pomocą przetwornika ADC, zbierający od niego dane i umieszczający je w buforze. Zawarty tam kod odpowiada także za multipleksowe sterowanie matrycą diod LED oraz za cykliczny odczyt stanu zwór J1 i J2. Ładowanie danych do bufora jest konieczne, bo czas przeliczania danych jest różny i niekiedy znacznie przekracza okres wywoływania przerwania. Zastosowanie bufora o długości aż 256 próbek sprawia, że wystarczy, aby średni czas przeliczania był mniejszy niż czas dostępny pomiędzy wyjściem z obsługi przerwania a ponownym wejściem w obsługę kolejnego przerwania. Wtedy program wyrobi się z obliczeniami i nie zgubi żadnej próbki. Różny czas obliczeń wynika stąd, że niektóre



Rys. 5. Skala 50%



Rys. 6. Skala 50%

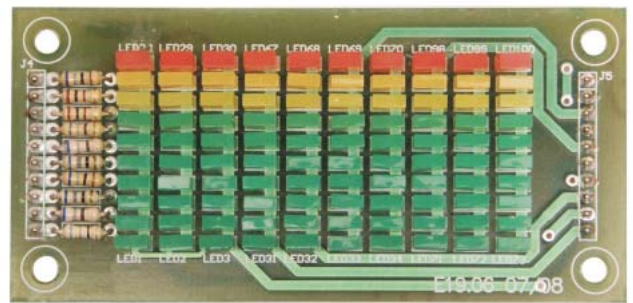
ment, kiedy trzeba jednocześnie uruchomić kilka z nich (czerwone kreski na rysunku 4). Filtrowanie jest dokonywane w programie głównym.

Przerwanie od Timera2 występuje co około 25 \square s (dokładnie co 24.889 \square s), co daje wspomnianą wcześniej częstotliwość próbkowania równą 40178Hz.

Dobroć filtrów wynosi około 6, a częstotliwości środkowe to 31Hz, 62Hz, 125Hz, 250Hz, 500Hz, 1kHz, 2kHz, 4kHz, 8kHz i 16kHz. Wartości amplitud, przy których zaświecają się kolejne diody, to -16dB, -14dB, ..., -2dB, 0dB i 2dB (co 2dB). Wskazanie 0dB odpowiada amplitudzie na wejściu analizatora równej około 0,2V.

Jumper J3 służy do wyboru trybu słupkowy/punktowy, a jumper J4 pozwala włączyć/wyłączyć wskaźniki maksymalnej wartości amplitudy. Jumpery wkłada się poziomo, tak jak widać na zdjęciach modelu.

Niestety dokładniejsze przedstawienie i wyjaśnienie całego programu jest bardzo trudne, bo jak już wcześniej wspominałem, jest on



filtry są uruchamiane rzadziej, ale co jakiś czas wystąpi moment,

napisany w assemblerze. Kto chce włączyć się w zawoilości programu, może to zrobić, ściągając kod źródłowy ze strony internetowej EdW. Podobnie jest ze wzorami i zasadami projektowania filtrów IIR. Jeżeli to mocno zainteresuje Czytelników EdW, być może wtedy pojawi się cykl artykułów poświęcony tej tematyce.

Montaż i uruchomienie

Układ można zmontować na 2 płytkach drukowanych przedstawionych na rysunkach 5 i 6. Płytki są składane w tzw. „kanapkę”. Montaż jest raczej klasyczny, poza tym że elementy są dosyć ciasno rozmieszczone. Na razie nie montujemy rezystorów R21...R30. Montaż rzadziej rozpocząć od elementów najmniejszych. Jedynym wyjątkiem jest układ U3, który trzeba wlutować przed zamontowaniem kondensatora C5. Gdybyśmy najpierw wlutowali kondensator C5, to późniejsze wlutowanie układu U3 mogłoby być niemożliwe. Sytuacja ta dotyczy tylko pionowego montażu tego kondensatora. Wyjaśnię to później. Pod układ U3 można dać podstawkę. Rezonator X1 montujemy na stojąco jedynie w przypadku, gdy będzie on miał wysokość nie większą niż 8mm. Jeżeli będzie to najpopularniejszy rodzaj rezonatora, czyli

Wykaz elementów

Płyta główna:

Rezystory

R1-R10, R34	4,7k Ω
R11-R20	470 Ω
R31, R33	68k Ω
R32, R35	6,8k Ω
R36, R37	47k Ω
R38 (nie montować)	100 Ω

Kondensatory

C1	470 μ F/16V
C2, C4, C6	100nF ceramiczny
C3	22 μ F
C5	100 μ F/6,3V
C7, C8	18pF
C9, C11	4,7 μ F
C10	10 μ F
C12	4,7nF ceramiczny

Półprzewodniki

D1-D4	1N4148
T1-T10	BC557

T11-T20	BC337
U1	7805
U2	TL071
U3	ATMega8 (16 DIL28)
U4	4028

Inne

Q1	kwarc 18MHz
J1	listwa goldpin 5x1
J2, J3	gniazdo na goldpiny 10x1
J4 i J5 jako jedno gniazdo	listwa goldpin kątowa 2x2 i dodatkowo 2 jumperki na to gniazdo

Wyświetlacz:

Rezystory

R21-R27	56 Ω
R28, R29	100 Ω
R30	68 Ω

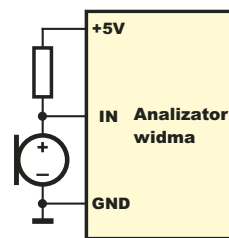
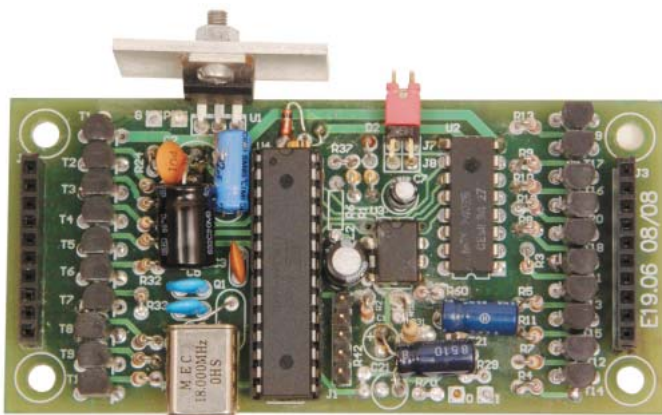
Półprzewodniki

D1-D10	LED 2x5mm czerwona
D11-D30	LED 2x5mm żółta
D31-D100	LED 2x5mm zielona

Inne

J6, J7	listwa goldpin 10x1
--------	---------------------

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2864.



Rys. 7

taki o wysokości 12mm, to należy go zamontować na leżąco, przechylając o 90° w stronę rezystorów R1...R10, tak jak jest to narysowane na płycie. Kondensatory elektrolityczne C1, C3, C9, C10 i C11 pionowo także mogą się nie zmieścić, więc je również należy zamontować na leżąco. Punkty lutownicze tych kondensatorów zostały tak rozmieszczone na płycie, że pionowy montaż przynajmniej trzech pierwszych z nich jest teraz bardzo utrudniony. Jeżeli nawet ktoś zdobędzie kondensatory o małej wysokości, to i tak będzie zmuszony zamontować je bokiem. Najwięcej kłopotów sprawi wspomniany wcześniej kondensator C5. Na szczęście jego napięcie pracy nie musi być wysokie i może wynosić na przykład tylko 6,3V. A kondensatory o tak małym napięciu pracy są zazwyczaj małe. Ten kondensator można zamontować pionowo jedynie, gdy będzie miał wysokość równą co najwyżej 8mm oraz odpowiednio małą średnicę. Jeżeli nie, to należy go położyć w kierunku miejsca na rezystor R38, który to rezystor, jak już wcześniej zaznaczyłem, nie będzie montowany. Aby ułatwić sobie taki boczny montaż kondensatora C5, kondensator C12 należy zamontować możliwie blisko układu U3. Układowi U2 przed jego zamontowaniem należy obciąć lub odgiąć do góry nóżki nr 1, 5 i 8, ponieważ płytka drukowana nie posiada na nie otworów. Listwy goldpinów (J2 i J3) na płycie wyświetlacza oraz gniazda na goldpiny (J6 i J7) na drugiej płycie najlepiej

ich w otworach na płytkach i skróceniu płytek. Dzięki temu zostaną one od razu zamontowane w miarę prosto i co najważniejsze, zamontowane, tak że płytki będą od razu idealnie pasowały do siebie. Listwy goldpinów J2 i J3 montujemy od strony druku. Tranzystory T1...T20 należy zamontować możliwie blisko płytki drukowanej tak, żeby nie wystawały powyżej gniazd na goldpiny. W ogóle żadne elementy nie powinny wystawać powyżej tych gniazd. Stabilizator U1 na pewno nie zmieści się pomiędzy płytki, jeżeli będzie wylutowany prosto w swoje punkty lutownicze. Należy go zamontować tak, jak to widać na zdjęciach modelu. Takie zamontowanie umożliwi przy okazji łatwe przykręcenie radiatora. Prototyp analizatora przechodził liczne modyfikacje i dlatego finalna płytka drukowana różni się od tej, którą widać na zdjęciach modelu. Dlatego nie należy bezwzględnie wzorować się na zdjęciach. Niektóre elementy w prototypie znajdują się pod płytką (po stronie lutowania) i dlatego nie widać ich na fotografiach. Układ połączeń jest oczywiście taki sam. Jeszcze tylko kilka zdań o diodach LED1...LED100. Aby je w miarę prosto zamontować, trzeba użyć powszechnie znanej sztuczki. Polega ona na włożeniu nóżek diod w odpowiednie otwory, odwróceniu płytki do góry nogami (diody na dół, punkty lutownicze na górę) i położeniu tak odwróconej płytki na jakimś płaskim podłożu, na przykład stole (oczywiście na czas odwracania trzeba czymś

zabezpieczyć diody przed wypadnięciem, na przykład kartką papieru, którą wyjmemy, gdy już położymy płytkę). Dzięki temu diody będą w stałej odległości od płytki. Teraz lutujemy po jednej nóżce każdej diody. Prylutowane na początku nóżki najlepiej jest od razu skracać, aby ułatwić sobie lutowanie kolejnych nóżek. Po przylutowaniu wszystkich potrzebnych nóżek odwracamy

płytkę i ustawiamy diody prosto, bo z pewnością będą trochę poprzekęcane i poprzechylane. Gdy już tego dokonamy, możemy przylutować „drugie” nóżki każdej diody. Układ po zmontowaniu zasadniczo nie wymaga uruchamiania i od razu powinien pracować poprawnie. Jednak przed ostatecznym podłączeniem sygnału audio podłączamy do niego szum różowy o wartości około 1Vpp. Taki sygnał spowoduje przesterowanie, ale o to nam w tym momencie chodzi. Zasilamy układ napięciem około 12V i wydajności co najmniej 300mA. I teraz niespodzianka. Nic nie zobaczymy na wyświetlaczu, bo nie wylutowaliśmy rezystorów R21...R30. Celowo jednak napisałem, żeby ich nie montować. Podczas budowy prototypu przekonałem się, że jasność diod LED różnych kolorów albo nawet tego samego koloru, ale różnych producentów, może się tak bardzo różnić, że wymagane jest staranne dobieranie wartości rezystorów, aby diody świeciły z w miarę zbliżoną jasnością.

Otwory w płycie są metalizowane i już samo włożenie nóżek rezystora w odpowiednie otwory powinno spowodować zaświecenie odpowiedniego rzędu diod.

Przygotowujemy sobie więc kilka...kilkanaście rezystorów z zakresu 33Ω...330Ω i wkładamy je kolejno w miejsce rezystorów R27 (rząd diod zielonych), R28 (żółtych) i R30 (czerwonych). Najczęściej diody mają prąd nominalny 20...30mA, ale w tym przypadku są sterowane ze współczynnikiem wypełnienia równym 10% i to z dużą częstotliwością, więc powinny wytrzymać prąd impulsowy około 40mA. Taki prąd uzyskamy stosując rezystory o wartości mniej więcej 33Ω. Rezystorów o mniejszej wartości nie powinniśmy używać. Gdy już dobierzemy rezystory tak, żeby diody świeciły ze zbliżoną jasnością i jednocześnie w miarę jasno, to możemy je wylutować na stałe. Dla przykładu w prototypie diody tak bardzo różniły się jakością, że aby uzyskać jako tako zbliżoną jasność, musiałem dla diod zielonych zastosować rezystory 33Ω, a dla żółtych aż 330Ω!

Teraz cały proces budowy mamy już za sobą. Układ powinien być podczas normalnej pracy sterowany sygnałem audio o amplitudzie do około 0,2Vpp. Wkładamy lub nie wkładamy jumperów J3 i J4 w celu włączenia odpowiedniego trybu pracy i to wszystko!

Do analizatora można także podłączyć mikrofon elektretowy, jak to przedstawiono na rysunku 7. W takim przypadku należy znacznie zwiększyć wzmocnienie wzmacniacza, zmieniając wartości przynajmniej jednego z rezystorów R32 i R33 (na przykład R32 na 4,7kΩ i R33 na 220kΩ). Jeżeli wartość rezystora R32 ma być mniejsza niż 6,8kΩ, to pojemność kondensatora C10 powinna odpowiednio wzrosnąć, aby nie tłumić sygnałów o najniższych częstotliwościach.

Andrzej Jabłoński
atom1477@wp.pl

R E K L A M A

RFID STEROWNIKI CZYTNIKI

www.mikrokontrola.pl
mikrokontrola

ul. Wólczyńska 55, 01-908 Warszawa, tel.: 0-22/ 865 55 45
fax: 0-22/ 865 55 44, e-mail: biuro@mikrokontrola.pl