

# Ethernetowy system audio

Jak wszyscy wiemy, stajemy się coraz bardziej leniwi i wygodnicy. Chcemy mieć najłatwiejszy dostęp do wszelkich multimediów. Dlatego tak prędko rozwija się komunikacja bezprzewodowa. Chcemy też uczynić swoje komputery jak najbardziej przenośnymi. W związku z tym coraz popularniejsze stają się interfejsy bezprzewodowe, takie jak WLAN czy Bluetooth. Dziwić może jedynie fakt, iż nie ma na rynku bezprzewodowego standardu transmisji dźwięku. Oczywiście telewizja cyfrowa nowej generacji (HD) posiada opracowany już standard – WirelessHD, jednak w praktyce użytkownik skazany jest na kolejne metry kabla audio. Prezentowany układ jest próbą pośredniego rozwiązania problemu i wykorzystuje fakt, że nawet w wielu domach przydatnych rozciągnięta jest sieć komputerowa Ethernet. *Ethernetowy system audio* działa jako przetwornik sygnału przesyłanego przez sieć Ethernet na sygnał analogowy audio. Proponowane eksperymentalne urządzenie jest wprawdzie mniej doskonałe niż rozwiązania fabryczne, choćby dlatego, że zapewnia jedynie 8-bitowe przetwarzanie, jednak przeszło domowe testy ku zadowoleniu Autorów. Poza tym jest tanie, a kolejnym atutem projektu jest satysfakcja z wykonania takiego nietypowego układu.

Urządzenie wykorzystuje typowe dla transmisji danych przez sieć standardowe protokoły:

1. Protokół IP (Internet Protocol) – protokół komunikacyjny warstwy sieci w stosie protokołów TCP/IP używany powszechnie w Internecie i sieciach lokalnych, i zapewniający bezpołączeniowe usługi sieciowe. Nie zapewnia niezawodności transmisji, którą dostarczają protokoły warstw wyższych.
2. Adres MAC (Media Access Control) – sprzętowy adres urządzenia sieciowego w sieci Ethernet.
3. Protokół ARP (Address Resolution Protocol) – protokół internetowy stosowany do przyporządkowania adresom IP adresów MAC.

4. Protokół UDP (User Datagram Protocol) – prosty protokół bezpołączeniowy w warstwie transportu odpowiadający za wymianę datagramów bez potwierdzania lub gwarancji dostarczenia.

5. Protokół ICMP (Internet Control Message Protocol) – protokół internetowy warstwy sieci, używany w diagnostyce sieci oraz routingu. Ważny w tym przypadku jest program sondujący Ping, który jest echem ICMP komunikatu i odpowiedzi, i stosowany jest do testowania dostępności urządzenia sieciowego.

## Opis układu

Nasze urządzenie podzielone jest na trzy moduły. Dzięki rozdzieleniu na moduły, zredukowany jest też problem zakłóceń.

Pierwszym modulem urządzenia jest interfejs internetowy, którego schemat ideowy jest przedstawiony na **rysunku 1**. Podstawowym elementem tego modułu jest układ ENC28J60 (U1) firmy DALLAS. Jest to samodzielny kontroler ethernetowy z popularnym interfejsem SPI (Serial Peripheral Interface). Interfejs SPI służy do dwukierunkowej (full-duplex), synchronicznej, szeregowej transmisji danych pomiędzy mikrokontrolerem a układami peryferyjnymi. Jest to interfejs trójżyłowy. Składa się z dwóch linii synchronicznie przesyłających dane w przeciwnych kierunkach (SI, SO) oraz linii z sygnałem zegarowym synchronizującym ten transfer (SCK). Za obsługę SPI w przypadku układu ENC28J60 odpowiadają piny 6, 7, 8, 9, czyli odpowiednio SO, SI, SCK i /CS. Ponadto układ wyposażony jest we własny, przypisywany mu adres MAC oraz, co ważne, ma zaimplementowaną warstwę fizyczną (PHY – Physical Layer). Poza tym ma również możliwość transmisji i odbioru danych. Całość zamknięta jest w dużej obudowie typu PDIP 28.

Układowi towarzyszy rezonator kwarcowy 25MHz. Rezystory pullup R10 i R1 profilaktycznie podciągają nam linie RESET i CS do zasilania. Pin RBIAS (będący pinem dla prądu polaryzującego warstwę fizyczną PHY) musi być ściągnięty do masy przez rezystor pull-

down R5. W układzie konieczny jest typowy transformator do sieci ethernetowej 10base-T, z przekładnią równą 1. Najwygodniejszym rozwiązaniem jest użycie złączki ethernetowej typu RJ-45 z wbudowanym transformatorem (MAGJACK). Dodatkowo mamy tu wbudowane dwie diody LED, sygnalizujące połączenie oraz transmisję danych. Potrzebna jest również mała cewka. Do tego celu nadaje się np. koralik ferrytowy 5mm z 5–7 zwojami cienkiego drutu, bądź gotowy dławik o indukcyjności 10µH. Moduł zasilany jest napięciem 3,3V. W tym celu potrzebny nam był stabilizator dodatkiego napięcia. Nasz wybór padł na układ LD1117V33.

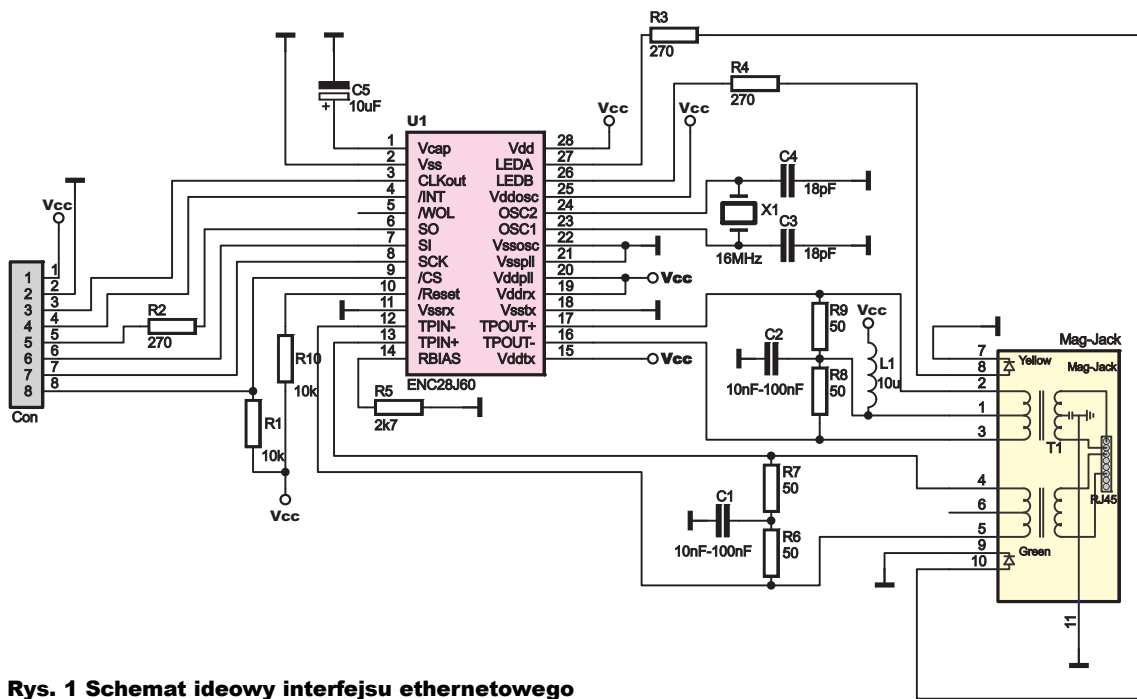
Drugim, najważniejszym modulem jest jednostka centralna, z „mózgiem” w postaci mikrokontrolera ATMEGA32 (U1), przedstawiona na **rysunku 2**. Do jego zadań należy sprawdzanie pakietów, następnie ich przetwarzanie, aż wreszcie wysyłanie dalej, na filtr RC i wzmacniacz. Zostały zaimplementowane dwa przetworniki cyfrowo-analogowe, ponieważ przetwarzany jest sygnał stereo. Aby znacząco zmniejszyć koszt projektu, przetworniki zostały zaimplementowane jako generatory PWM z filtrami dolnoprzepustowymi. Wadą tego rozwiązania jest możliwość operowania jedynie na 8 bitach. PWM (Pulse Width Modulation) to metoda regulacji sygnału prądowego lub napięciowego, polegająca na zmianie szerokości impulsu o stałej amplitudzie. Dzięki odpowiedniej regulacji średniej wartości impulsu prostokątnego, po uśrednieniu otrzymujemy pożądaną wartość analogowego napięcia wyjściowego. Prosty sposób zobrazowaniem działania generatora PWM jest **rysunek 3**. Inną bardzo ważną rolą, jaką pełni nasze „jądro” systemu, jest sterowanie pamięcią RAM. Układ taktowany jest z rezonatora kwarcowego 16MHz.

Jako pamięć RAM wykorzystaliśmy układ BS62LV1027 (U3). Jest to pamięć SRAM 8 x 128kb, przechowująca nasze próbki, dzięki czemu transmisja się nie zacina. Pamięć SRAM zawiera układ kontrolny, sterowany sygnałami CE1 (Chip Enable), CE2,

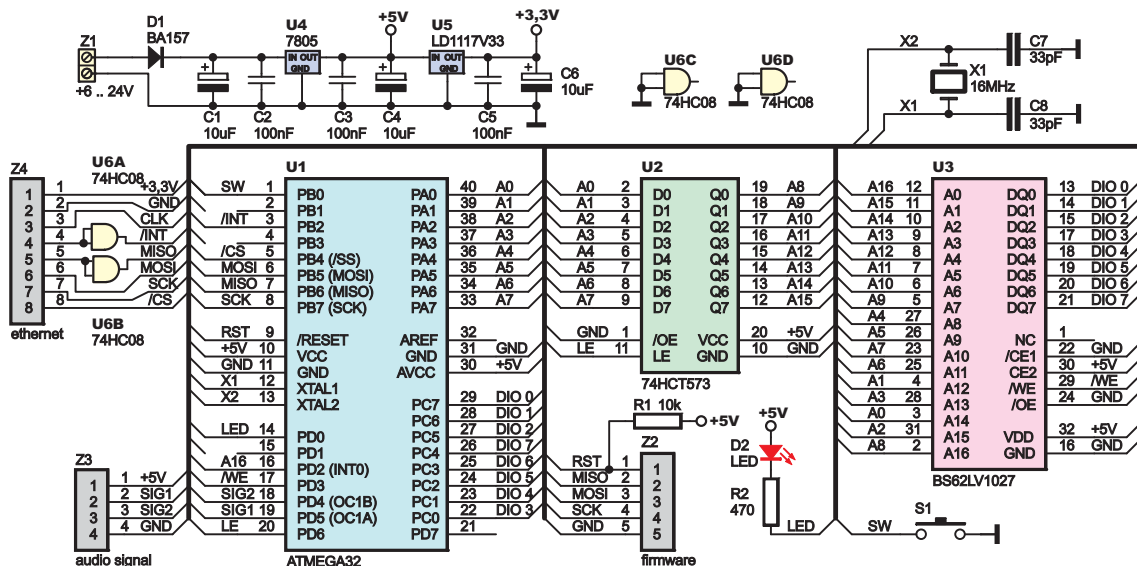
WE (Write Enable) i OE (Output Enable). Wybór komórki (bajtu) do zapisu lub odczytu odbywa się za pomocą linii adresowych A0-A16. Sygnały CE1 i CE2 służą do uaktywnienia układu. Sygnał OE jest wykorzystywany w trakcie czytania danej, a sygnał WE w trakcie pisanie danej do pamięci. Układ 74HCT573 (U2) zawierający 8 buforów trójstanowych oraz 8 przerzutników typu D pracuje tu w typowy dla siebie sposób i zmniejsza liczbę linii adresowych. Najpierw z procesora do przerzutników U2 przez linie A0...A7 ładowana jest i zapamiętywana starsza część bitów adresu. W kolejnym cyklu pamięć adresuje młodszy bajt z linii A0...A7, zatrasnięta zawartość kostki U2 oraz linia A16.

Układy stabilizujące napięcie zasilające znajdują się właśnie w tym module. Pierwszy jest stabilizator dodatkowego napięcia 3,3V, potrzebnego do zasilania modułu interfejsu ethernetowego (LD1117V33 - U5). Drugim ważnym układem jest 7805 (U4). Napięcie 5V potrzebne jest do zasilania tego modułu, a przede wszystkim procesora. Ze względu na wydolność częstotliwościową, specjalnie dobrany został układ ATMEGA32, który potrzebuje właśnie takiego napięcia zasilającego, a nie np. ATMEGA32L. Napięciem 5V zasilamy również filtry RC oraz wzmacniacze do generatorów PWM.

Moduł wyposażony został także w bramkę AND – 74HC08 (U6) pełniącą rolę bufora. Bufor jest opcjonalny, niemniej warto go zastosować. Ponadto zaimplementowany został RESET – przycisk S1. W tej sytuacji następuje zaświecenie diody, co sygnalizuje kasowanie pamięci EEPROM procesora.



Rys. 1 Schemat ideowy interfejsu ethernetowego



Rys. 2 Schemat ideowy jednostki centralnej urządzenia



Rys. 3 Generator PWM i filtr dolnoprzepustowy jako przetwornik cyfrowo-analogowy

Trzecim modulem naszego projektu jest przedstawiona na rysunku 4 płytką zawierająca dwa filtry RC do generatora PWM oraz wzmacniacze. Jest to stopień wyjściowy urządzenia. Na wejściu obu kanałów mamy kaskadowe połączenie filtrów RC pierwszego rzędu, dobranych do częstotliwości 22kHz. Na wejście filtrów podawany jest sygnał z generatorów PWM z ATMEGA32. Następnym krokiem jest podanie sygnałów na wtórniki z układu TL072 (U1), zapew-

niające niską impedancję wyjściową. Finalnie na wyjściach Z1...Z3 otrzymujemy pożądany sygnał analogowy.

## Opis programu

Program dla mikrokontrolera został napisany w języku C. Warto byłoby zaznaczyć, iż posiadamy gotowy sterownik do układu ENC28J60, który jest dostępny na licencji GPL V2. Poza tym mamy również dostęp do sterowników obsługi protokołów IP, ARP i UDP. Pozwala nam to na swobodne używanie tego oprogramowania bez zagrożenia złamania praw autorskich. Autorem jest Guido Socher, a wyżej wymienione oprogramowanie jest dostępne na następującej stronie internetowej:

<http://tuxgraphics.org/electronics/200611/article06111.shtml>

Pliki na tej licencji to: *ENC28j60.c* oraz *IP\_ARP\_UDP.c* wraz z plikami nagłówkowymi o takich samych nazwach.

Program rozpoczyna się od inicjalizacji kolejnych modułów procesora ATMEGA32. Listing tej części znajduje się na rysunku 4. Funkcja *timer0int()*; nam timer. Funkcja *PWMint()*; ini-

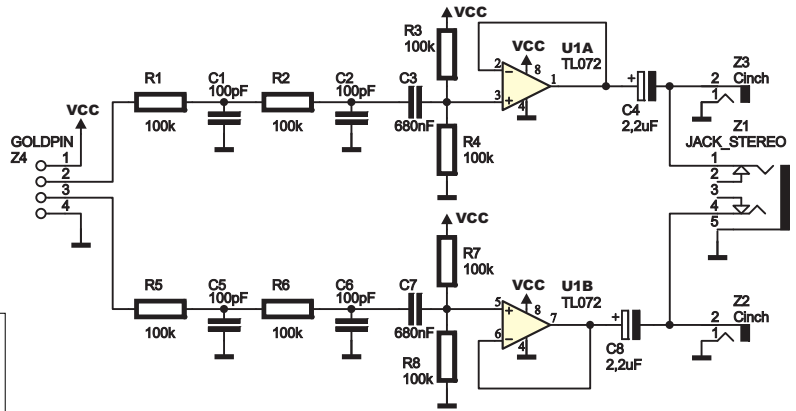
cializuje generator PWM. Jest to przeprowadzone poprzez zapis odpowiednich wartości do rejestrów procesora. Następnie inicjalizujemy kolejkę FIFO za pomocą funkcji

### Listing 1

```

while(1){
  reset_device(); // get the next new packet:
  plen = enc28j60PacketReceive(BUFFER_SIZE, buf);
  //plen will ne unequal to zero if there is a valid
  if(plen==0)
  {
    continue;
  }
  wdt_enable(WDTO_1S);
  wdt_reset(); //arp is broadcast if unknown but a
  // host may also verify the mac address by sending
  // it to a unicast address.
  if(eth_type_is_arp_and_my_ip(buf,plen)
  {
    make_arp_answer_from_request(buf,plen);
    continue;
  }
  // check if ip packets (icmp or udp) are for us:
  if(eth_type_is_ip_and_my_ip(buf,plen)==0)
  {
    continue;
  }
  if(buf[IP_PROTO_P]==IP_PROTO_ICMP_V →
    → && buf[ICMP_TYPE_P]==ICMP_TYPE_ECHOREQUEST_V)
  {
    // a ping packet, let's send pong
    make_echo_reply_from_request(buf,plen);
    continue;
  }
  // odczyt ramki z próbkami
  if (buf[IP_PROTO_P]==IP_PROTO_UDP_V →
    → &&buf[UDP_DST_PORT_H_P]==4 →
    → &&buf[UDP_DST_PORT_L_P]==0xb0)
  {
    payloadlen=buf[UDP_LEN_L_P]-UDP_HEADER_LEN;
    for(i=42;i<200+42;i+=2)
    {
      if ( i == 44 )
      {
        usage = usage_FIFO();
        if(usage <= limes_low)
        {
          INTERRUPT IS COMMING();
          write_FIFO(buf[i],buf[i+1]);
          INTERRUPT IS COMMING();
          write_FIFO(buf[i],buf[i+1]);
        }
      }
      if(( usage < limes_high ) &&( usage > limes_low ))
      {
        INTERRUPT IS COMMING();
        write_FIFO(buf[i],buf[i+1]);
      }
    }
  }
  else
  {
    INTERRUPT IS COMMING();
    write_FIFO(buf[i],buf[i+1]);
  }
}
continue;
}
// konfiguracja IP i MAC
if (buf[IP_PROTO_P]==IP_PROTO_UDP_V →
  → &&buf[UDP_DST_PORT_H_P]==0x0D →
  → &&buf[UDP_DST_PORT_L_P]==0x4D)
{
  EEPROM_write(0,0x00);
  for (i=1; i < 11; i++)
  {
    EEPROM_write(i,buf[i+41]);
  }
}
continue;
}

```



**Rys. 4 Schemat ideowy filtrów RC i wzmacniaczy do generatora PWM**

*init\_FIFO()*. Krokiem jest wyłączenie watchdoga: *wdt\_off()*; tym musimy skonfigurować adresy IP oraz MAC naszego urządzenia. Do tego celu posłuży nam funkcja *configure\_IP\_MAC()*. Funkcja ta na początku przeprowadza próbę odczytu adresów IP i MAC z pamięci EEPROM procesora. Jeśli zakończy się to niepowodzeniem, domyślny adres IP urządzenia zostanie ustawiony na 10.0.1.1. Finalnie inicjalizujemy układ ENC28j60 za pomocą funkcji *enc28j60Init(mymac)*; tym samym wyjścia procesora tak, aby funkcjonował interfejs SPI.

Następnie omówimy funkcję *main()*; i znajdującą się w niej nieskończoną pętlę główną programu. Listing tej części programu to **listing 1**. Rozpocznijmy od funkcji przeprowadzającej reset urządzenia, *reset\_device()*; przeprowadzana jest ona następująco: najpierw sprawdzamy, czy nie naciśnięto przycisku reset. Jeśli ów przycisk został naciśnięty, oczekujemy odpowiednią ilość czasu (około 7 sekund), po czym

następuje ponowne sprawdzenie przyciśnięcia przycisku. Potem trzykrotnie zaświeca się dioda, a pamięć EEPROM jest kasowana. Jest to równoznaczne z powrotem do ustawień domyślnych. Ponieważ procesor nie może zostać zresetowany w sposób programowy, reset urządzenia jest przeprowadzany za pomocą watchdoga. Kolejną funkcją to *enc28j60PacketReceive(BUFFER\_SIZE, buf)*; Za jej pomocą dokonywane jest sprawdzenie czy układ ENC28j60 nie odebrał nowego pakietu internetowego. Funkcja zwraca wielkość tego pakietu. Jeśli otrzymamy 0, wracamy na początek pętli głównej. Kolejnym krokiem jest reset watchdoga za pomocą funkcji *wdt\_reset()*. Potem następuje sprawdzenie, czy pakiet nie jest pakietem ARP. Jeśli jest pakietem ARP, generujemy odpowiedź i wracamy na początek pętli głównej. Jeśli pakiet nie był pakietem ARP sprawdzamy, czy adres IP wysłanej ramki zgadza się z IP naszego urządzenia. Jeśli tak nie jest, to wracamy do początku pętli głównej. Następnie sprawdzamy, czy pakiet nie jest pakietem ICMP typu EchoRequest (procedura pingowania). Jeśli nim jest, wysyłana jest odpowiedź oraz następuje powrót na początek pętli głównej. Potem sprawdzamy, czy ramka jest ramką UDP oraz czy port się zgadza (ramka z dźwiękiem wysyłana jest zawsze na port 1200). Jeśli tak, następuje poprzedzony procedurą synchronizacji zapis do bufora FIFO. Synchronizacja jest bardzo potrzebna, gdyż częstotliwość procesora po podzieleniu przez timer nie jest idealnie równa pożądanej wartości. Bierzymy w tym przypadku pod uwagę dwie sytuacje. Pierwsza ma miejsce, gdy bufor jest bliski wyzerowania. Jest to równoznaczne ze zbyt małą ilością danych. W tym przypadku pierwsza próbka w

**Tabela 1 Ramka IP UDP**

8 bajtów		6 bajtów		6 bajtów		2 bajty		24 bajty		8 bajtów		4 bajty			
PREAMBUŁA		ADR. MAC ODB.		ADR. MAC NAD.		TYP/DŁUGOŚĆ		NAGŁÓWEK IP		NAGŁ. UDP		DANE		CRC	
Ramka IP UDP															
0	4	8	12	16	20	24	28	32							
WERSJA		DŁUGOŚĆ NAGŁ.		RODZAJ USŁUGI		FLAGI		DŁUGOŚĆ PAKIETU							
CZAS ŻYCIA		IDENTYFIKATOR		PROTOKÓŁ		SUMA KONTROLNA NAGŁÓWKI		PRZESUNIĘCIE PAKIETU							
				ADRES IP NADAWCY				ADRES IP ODBIORCY							
				OPCJE				UZUPEŁNIENIE							
PORT UDP NADAWCY				PORT UDP ODBIORCY				SUMA KONTROLNA UDP							
DŁUGOŚĆ KOMUNIKATU UDP															

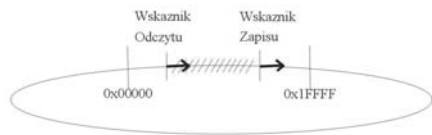


każdym pakiecie jest podawjana (to znaczy, że co 200 próbka jest wpisywana dwukrotnie). Druga sytuacja ma miejsce, gdy bufor jest bliski przepełnieniu. Wtedy pierwsza próbka jest pomijana.

W wypadku gdyby port, na który został wysłany pakiet, nie był portem 1200, następuje ponowne sprawdzenie numeru portu. Sprawdzane jest, czy pakiet nie jest pakietem konfiguracyjnym (port 1377 – odpowiada ramce konfiguracyjnej). Jeśli odebrana ramka zawiera nową konfigurację urządzenia, wtedy następuje zapis nowego adresu IP i MAC do pamięci EEPROM. Na koniec następuje powrót do początku pętli głównej i odczyt kolejnej ramki.

Warto napisać również parę słów wyjaśnienia, co do zaimplementowanej kolejki FIFO. Ogółem FIFO polega na tym, że pierwsza dana, które wejdzie do kolejki, będzie pierwszą, która z niej wyjdzie (First In First Out). Programowo bufor ten jest implementowany następująco. Istnieją dwa wskaźniki, które wskazują nam adres zapisu i odczytu z zewnętrznej pamięci SRAM. Jeśli następuje zapis, inkrementowany jest wskaźnik zapisu, i odwrotnie, jeśli następuje odczyt, inkrementowany jest wskaźnik odczytu. Obrazuje to rysunek 5. Na koniec, w tabeli 1 możemy zobaczyć wygląd ramki IP UDP.

Poza oprogramowaniem wgrzywanym na procesor ATMEGA32, potrzebny jest również program na komputerze wysyłający próbki przez sieć Ethernet. Dla ułatwienia został zamieszczony kod na dwóch platformach: Java oraz Delphi. Zasada działania ogranicza się do następujących czynności. Dźwięk jest nagrywany. Oznacza to, że program pracuje w konfiguracji recordera. To, co zostało nagrane, nie jest jednak zapisywane na dysk. Jest natomiast wysyłane przez Internet.

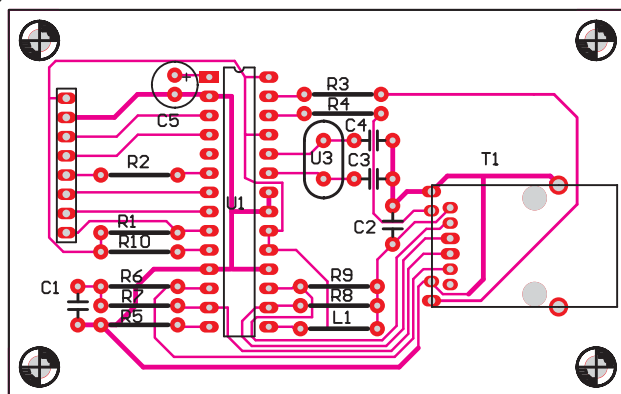
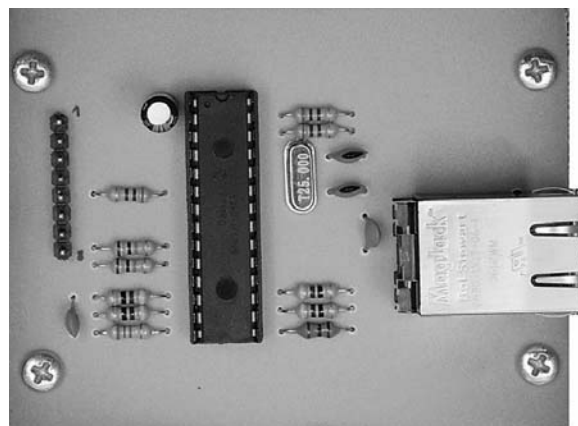


Rys. 5 Kolejka FIFO

## Montaż i uruchomienie

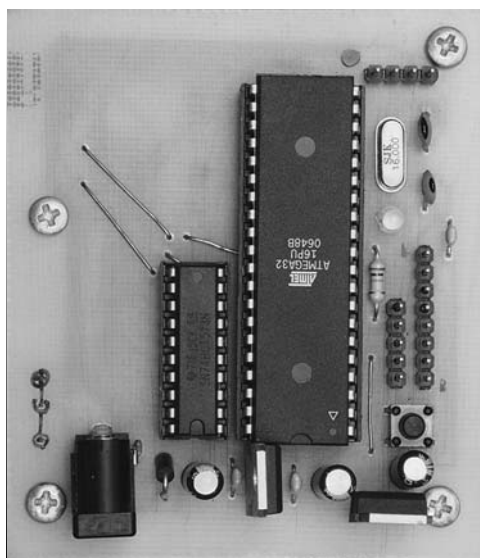
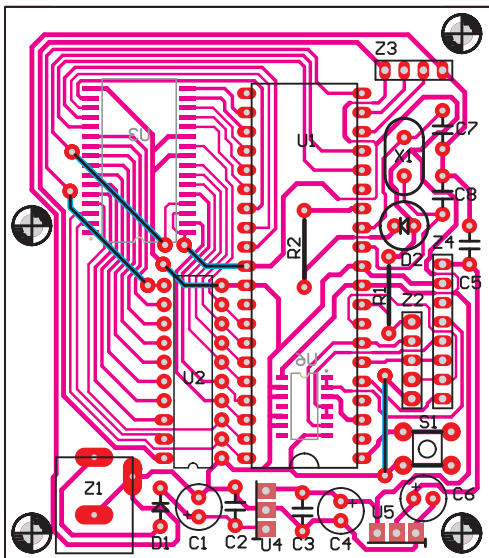
W wyniku podziału naszego ethernetowego systemu muzycznego na moduły, każda płytką została zaprojektowana oddzielnie. Druk na każdej z nich jest jednostronny (jedynie przy jednostce centralnej występują zworki). Wzory płytek można zobaczyć kolejno: rysunek 6 – interfejs ethernetowy, rysunek 7 – jednostka centralna, rysunek 8 – stopień wyjściowy.

Sam montaż nie powinien przysparzać trudności. Najpierw wlotujemy elementy najniższe, czyli elementy SMD, zworki, rezystory, cewki i diody. Należy zwrócić uwagę, że niektóre zworki mają wspólne otwory z nóżkami układów scalonych! Następnie montujemy podstawki pod układu scalone, kondensatory, rezonatory kwarcowe oraz stabilizatory. Na koniec zajmujemy się montażem wszelkich złączy znajdujących się w układzie, zaczynając od najmniejszych. Zastosowaliśmy dwa standardy złączy wyjściowych z układu (płytką z filtrem i wzmacniaczem): 2 x cinch (na lewy i prawy kanał) oraz jack stereo. Dzięki temu urządzenie zyskuje na funkcjonalności. Konektory oznaczone na rysunkach jako Z to goldpiny.



Rys. 6 Schemat płytki montażowej interfejsu ethernetowego

Rys. 7 Schemat płytki montażowej jednostki centralnej



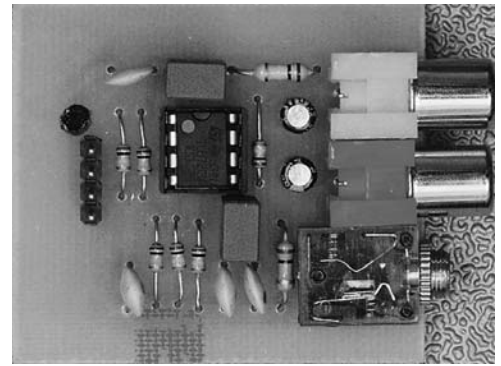
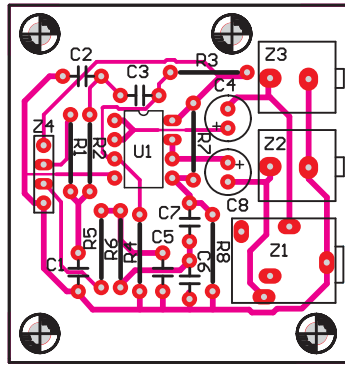
Złącze Z2 na płytce z jednostką centralną daje nam możliwość programowania mikrokontrolera ATMEGA32 w systemie (ISP – In System Programmable). W mikrokontrolerze Atmega32 domyślnie włączona jest obsługa interfejsu JTAG. Z uwagi na to, iż używa on pinów, do których podłączamy pamięć SRAM, ważne jest, żeby bit konfiguracyjny JTAGEN był zaprogramowany (bit = 0). Wtedy JTAG będzie wyłączony, a piny 24, 25, 26, 27 będą służyć jako normalny port C. Ustawienie bitów konfiguracyjnych w programie PonyProg pokazane jest na rysunku 9. Układ zasilany może być napięciem stałym od 6V do 24V.

Po zmontowaniu i podłączeniu układu do prądu urządzenie nie da znaku życia. Dopiero po podłączeniu do działającej sieci internetowej zaświeci się zielona dioda w złączu MagJack. Urządzenie będzie gotowe do pracy. Jednakże, w większości przypadków, konieczne będzie przeprowadzenie konfiguracji adresu IP.

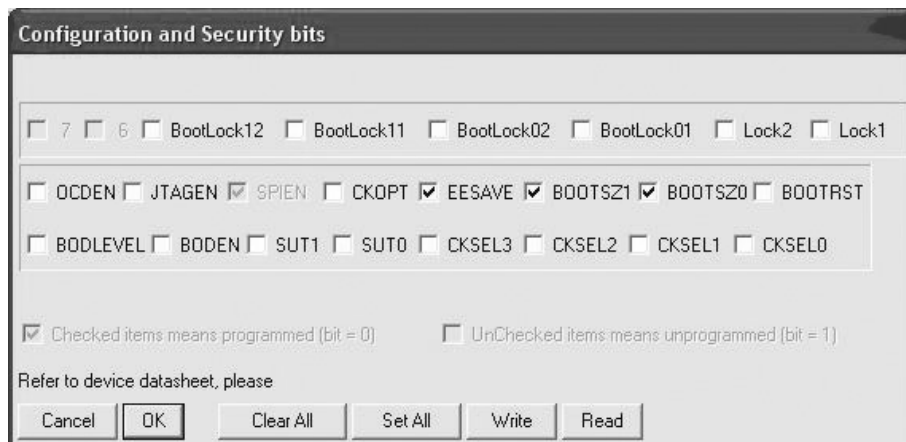
Domyślnym adresem IP urządzenia jest 10.0.1.1. Są trzy metody zmiany adresu IP. Pierwsza (najprostsza) to użycie programu w Delphi. Druga to zmiana domyślnego numeru IP w kodzie źródłowym. Trzecia to ręczna edycja pamięci EEPROM procesora ATMEGA32. Przykładowy wpis do pamięci jest następujący: 0x00, 0x0A, 0x00, 0x01, 0x01, 0x0A, 0x0B, 0x0C,

0x0D, 0x0E, 0x0F. Pierwszy bajt musi zawierać 0. Pozostałe bajty konfigurują numer IP na 10.0.1.1 i MAC na 0A, 0B, 0C, 0D, 0E, 0F.

Komputerowy program napisany w Delphi jest aplikacją bardzo przyjazną użytkownikowi. Posiada graficzny interfejs użytkownika, a każda kontrolka jest dokładnie opisana. Zawiera także obszernie opisy i podpowiedzi co do konfiguracji. Program napisany w Javie nie ma graficznego interfejsu użytkownika. Niekoniecznie musi być to wadą. Wystarczy uruchomić aplikację (można dodać skrót do Autostartu, aby aplikacja startowała przy każdym uruchomieniu systemu), po czym możemy zapomnieć o jej istnieniu. Ponadto program ten jest znacznie stabilniejszy, i z tej racji bar-



**Rys. 8 Schemat płytki montażowej filtrów i wzmacniaczy do generatora PWM**



**Rys. 9 Ustawienie bitów konfiguracyjnych w programie PonyProg**

dziej zaawansowanym użytkownikom zalecamy korzystanie właśnie z tej wersji.

Poza tym udostępniliśmy także **emulator układu**, dzięki któremu będzie można sprawdzić, jak działa fizyczne urządzenie, zanim cokolwiek zostanie polutowane. Obsługa emulatora sprowadza się do uruchomienia go na innym komputerze, który będzie „udawał” układ fizyczny. Program emulatora nie wymaga i nie posiada żadnej konfiguracji, działa od razu po włączeniu.

Całe oprogramowanie (tak na komputer, jak i te wgrywane na mikrokontroler ATMEGA) jest dostępne na stronie Elportalu.

Paweł Kniola pawel.kn@gmail.com

Krzysztof Woronowicz kwor@wp.pl

Wojciech Czapkowski blackhearted\_angel@o2.pl

## Wykaz elementów

<b>MAINBOARD</b>	C4, C8	.....	2,2µF
<b>Kondensatory</b>	<b>Rezystory</b>		
C1, C4, C6	R1-R8	.....	100kΩ
C2, C3, C5	<b>Półprzewodniki</b>	.....	100nF
C7, C8	U1	.....	33pF
<b>Rezystory</b>	<b>Pozostałe</b>		
R1	Z1	.....	10kΩ
R2	Z2, Z3	.....	470Ω
<b>Półprzewodniki</b>	Z4	.....	listwa goldpinów
D1	<b>INTERFEJS ETHERNETOWY</b>		
D2	<b>Kondensatory</b>		
U1	C1, C2	.....	10nF-100nF
U2	C3, C4	.....	18pF
U3	C5	.....	10µF
U4	<b>Rezystory</b>		
U5	R1, R10	.....	10kΩ
U6	R2, R3, R4	.....	270Ω
X1	R5	.....	2,7kΩ
<b>Pozostałe</b>	R6, R7, R8, R9	.....	50Ω
S1	<b>Półprzewodniki</b>		
Z1	U1	.....	ENC28J60
Z2-Z4	X1	.....	kwarc 25MHz
<b>FILTR R-C DO PWM</b>	<b>Pozostałe</b>		
<b>Kondensatory</b>	T1	.....	Magjack
C1, C2, C5, C6	Listwa goldpinów	.....	8 pinów
C3, C7	L1	.....	dtawik 10µH

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2877.