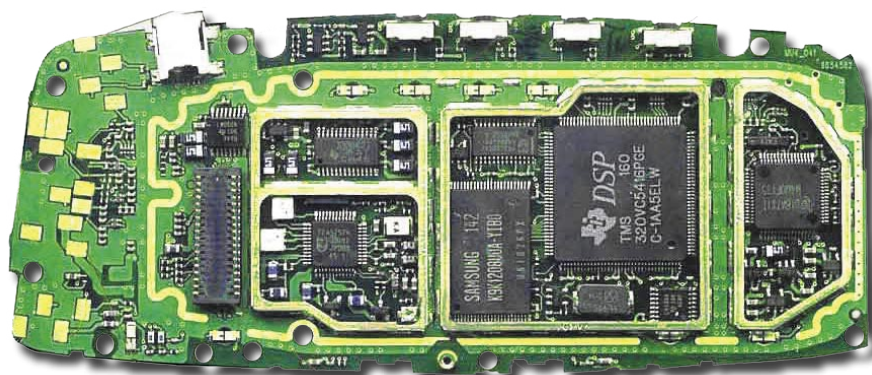




# TMS320VC5416 z Nokii 5510

## Zestaw ewaluacyjny dla procesora sygnałowego (2)

Przedstawiamy drugą część opisu wykonania zestawu ewaluacyjnego przeznaczonego dla procesorów DSP. Podano w nim informacje uzupełniające na temat montażu i uruchomienia płytki, a także bootloadera, układów pamięci i procesora DSP.



### Bootloader

Procesor TMS320VC5416 posiada wewnętrzną pamięć ROM, w której umieszczony jest *bootloader*. W telefonie Nokia 5510 ładuje on kod bezpośrednio z pamięci NAND Flash do RAM-u. Aby go uruchomić, należy podczas zerowania układu podać stan niski na wyjście MP!/MC. Wówczas wewnętrzny ROM jest mapowany w przestrzeni adresowej pamięci programu. Niestety, nie udało mi się znaleźć dokładniejszych informacji dotyczących działania tego *bootloadera*, dlatego nie pozostaje nic innego, jak opracować własny program kopiujący dane do wewnętrznej pamięci. Bezpośrednie wykonywanie rozkazów z pamięci NAND ze względu na jej budowę nie jest możliwe. Trzeba skorzystać z innego sposobu przekazywania instrukcji, mianowicie podczas zerowania DSP na linii BOOT mikrokontroler ustawia stan wysoki, sygnalizując tym samym

układowi logiki programowalnej, że od tego momentu wszelkie odczyty DSP z zewnętrznej pamięci dotyczyć będą procesu bootowania. Po wyzerowaniu TMS320VC5416 do licznika rozkazów PC ładowany jest adres 0xFF80. Jeśli na nóżce MP!/MC jest stan wysoki, to zamiast ROM-u mapowana jest w tym miejscu pamięć zewnętrzna. W momencie, gdy procesor odwoła się do niej w celu odczytania pierwszego rozkazu (sygnały sterujące !PS=!MSTRB=0 R!/W=1), CPLD ustawia na linii READY stan niski, wymuszając w ten sposób dodatkowe stany opóźnienia odczytu (rys. 11). Teraz inicjatywę przejmuje mikrokontroler MSP430F135. Sprawdza on istnienie pliku *boot.dat* w systemie FAT32 wewnątrz Flasha, a następnie zaczyna „podrzucać” procesorowi podczas kolejnych odczytów z pamięci, rozkazy zapisujące kolejne bajty pliku do pamięci RAM. Pamięć Flash posiada 8-bitowe wyjście danych, dlatego wymagane jest wykorzystanie zatrząsków wewnątrz układu logiki programowalnej. Wartości są zatrząskiwane parami, na opadających zboczach linii BOOT\_D, a następnie kierowane na 16-bitową szynę danych.

Sam format pliku *boot.dat* jest taki sam, jak w przypadku fabrycznego *bootloadera* Texas Instruments dla standardowego 8-bitowego trybu szeregowego (tab. 7) opisanego w dokumencie „TMS320VC5402A/VC5409A/VC5410A/VC5416 Bootloader”. Składa się on z bloków danych wraz z docelowym adresem i rozmiarem. W przypadku braku lub nieprawidłowości w strukturze *boot.dat*, błąd jest sygnalizowany migającą diodą D1.

Po transporcie wszystkich części danych mikrokontroler chcąc odczytać rozmiar kolejnego bloku natrafia na wartość 0000h, oznaczającą

>> część pierwsza tego artykułu dostępna jest na CD

## AVT-5174

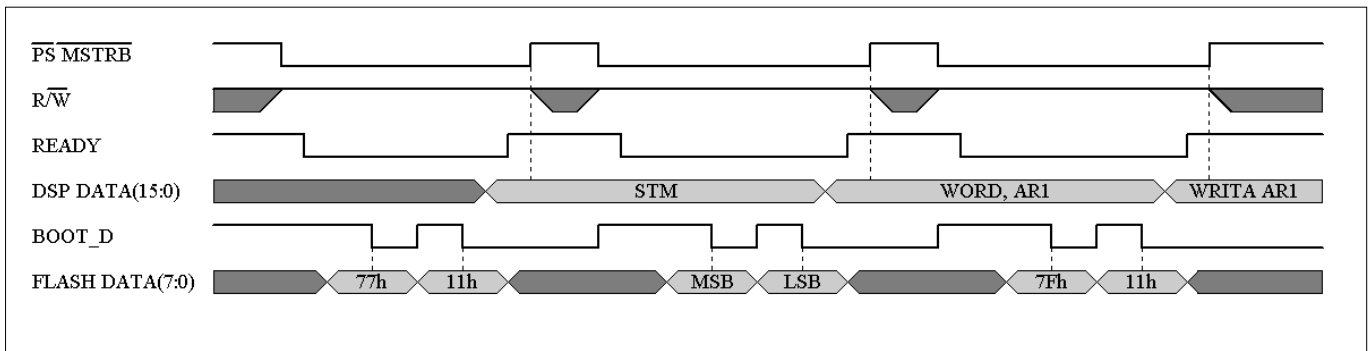
W ofercie AVT:  
AVT-5174A – płytka drukowana

### PODSTAWOWE PARAMETRY

- Płytko o wymiarach 119×98 mm
- Zasilanie 6...15 V
- Gniazda: PHONES, LINE OUT, LINE IN, MIC, USB, PS2, DSP JTAG, CPLD JTAG, MSP430 JTAG, wielokanałowy buforowany port szeregowy McBSP, interfejs równoległy HPI8 (adres i dane multipleksowane na 8-bitowym porcie)
- Pamięć SRAM: 512 kB×16
- Pamięć NAND Flash: 64 MB×8
- Wyświetlacz LCD: matryca 640×480 16-bit
- Częstotliwość próbkowania sygnałów audio: 8...96 kHz

>> PROJEKTY POKREWNE wymienione artykuły są w całości dostępne na CD

Tytuł artykułu	Nr EP/EdW	Kit
Zestaw startowy dla mikrokontrolerów PsoC	EP 4/2006	AVT-926
Zestaw startowy dla mikrokontrolerów ST7FLITE2x	EP 7-8/2006	AVT-939
Zestaw uruchomieniowy dla procesorów 89CX051 i AVR	EP 3/2000	AVT-854
Emulator procesorów 89CX051	EdW 3/2000	AVT-2501

Rys. 11. Zależności pomiędzy sygnałami podczas przekazywania rozkazów *bootloadera* do DSP

koniec pliku. Proces bootowania kończony jest ustawieniem stanu niskiego na linii BOOT. Od tego momentu wszystkie odczyty i zapisy do pamięci są swobodnie wykonywane przez DSP.

### PDIUSB12 i K9K1208U0A

Jak pokazano na schemacie blokowym (rys. 2), pomiędzy pamięcią Flash a DSP znajduje się układ logiki programowalnej. Pracuje on w roli dekodera adresu oraz pośredniczy

w wymianie danych. Pamięć K9K1208U0A ma jedną 8-bitową szynę, na której są multiplexowane dane, adres oraz komendy. Poprzez wejścia ALE i CLE możemy dokonać wyboru, do której z wymienionych lokalizacji chcemy mieć dostęp. Ze względu na potrzebę zapewnienia odpowiednich sekwencji sygnałów sterujących przy operacjach na pamięci Flash, DSP manipuluje nimi niezależnie poprzez zapisy do odpowiednich portów I/O. W tab. 8 przedstawiono

ich rozmieszczenie w przestrzeni adresowej I/O procesora i ich opis. W przypadku PDIUSB-D12 nie jest wymagany tak skomplikowany interfejs dostępu. Posiada on dwa porty. Poprzez pierwszy z nich dokonywany jest zapis lub odczyt danych do/z wewnętrznego bufora. Drugi port służy do konfiguracji układu i wysyłania komend.

### Interfejs LCD

Podobne konstrukcje oparte na logice programowalnej były już opisane na łamach czasopisma Elektronika Praktyczna, dlatego nie będę się zagłębiać w szczegółową zasadę działania wyświetlaczy i kontrolerów graficznych. W prezentowanym układzie wykorzystywana jest matryca 640x480 16-bit. Układ taktowany jest zegarem 40 MHz pochodzącym z generatora G1, następnie częstotliwość ta jest dzielona przez 2 i podawana do wyświetlacza (złącze CON7). Na płytce znajdują się dwie kości SRAM o organizacji 16-bitowej, których łączna pojemność wynosi 1 MB. Jeden piksel na ekranie odpowiada jednej lokalizacji w pamięci. Kolejno wyświetlane komórki pamięci tworzą obraz. Niestety w przestrzeni adresowej DSP nie ma wystarczająco dużo miejsca, aby zmieścić cały bufor wideo w sposób liniowy. Pamięć programu TMS320VC5416 jest podzielona na 128 stron po 64 k słów każda. Począwszy od strony o numerze 4, obszar o adresie XX8000h–XXFFFFh odwołuje się do pamięci zewnętrznej. Poprzez prostą translację adresów w układzie logiki programowalnej, SRAM został podzielony na sekcje po 32 k i umieszczony na każdej stronie. Widać to na rys. 12. Funkcje

Tab. 7. Format pliku *boot.dat*

Bajt	Wartość	Opis	
0	08h	Sygnatura	
1	AAh		
2	X		
3	X		
4	X		
5	X		
6	X		
7	X		
8	X		
9	X		
10	MSB	Numer strony początku programu, młodsze 7 bitów ładowane do XPC	
11	LSB		
12	MSB	Adres początku programu, ładowany do PC	
13	LSB		
14	MSB	Rozmiar (X) bloku nr 0 wyrażony w dwubajtowych słowach	Blok nr 0
15	LSB		
16	MSB	Numer docelowej strony bloku (młodsze 7 bitów)	
17	LSB		
18	MSB	Adres docelowy bloku	
19	LSB		
20	MSB	WORD[1]	
21	LSB		
20+2X-2	MSB	WORD[X]	
20+2X-1	LSB		
20+2X	MSB	Rozmiar (Y) bloku nr 1 wyrażony w dwubajtowych słowach	Blok nr 1
21+2X	LSB		
22+2X	MSB	Numer docelowej strony bloku (młodsze 7 bitów)	
23+2X	LSB		
24+2X	MSB	Adres docelowy bloku	
25+2X	LSB		
26+2X	MSB	WORD[1]	
27+2X	LSB		
26+2(X+Y)-2	MSB	WORD[Y]	
26+2(X+Y)-1	LSB		
2N-2	00h	Oznaczenie końca sekcji bloków	
2N-1	00h		



put\_pixel(), area\_fill(), print\_str() oraz area\_copy() w module graphics.h, pokazują w jaki sposób można się odwołać do pamięci wideo.

**Audio**

Kodek TLV320AIC23 umożliwia nagrywanie (mikrofon – J1, line in – J2) oraz odtwarzanie (line out – J3, słuchawki – J4) dźwięku w szerokim zakresie częstotliwości próbkowania od 8 do 96 kHz. Zapisem do jego rejestrów konfiguracyjnych, poprzez magistralę I<sup>2</sup>C, zajmuje się układ MSP430F135. Służy do tego funkcja AIC23write(). Gdy procesor chce zmienić jakąś wartość w rejestrze konfiguracji kodeka, musi powiadomić o tym mikrokontroler poprzez interfejs SPI. Przesyłanie strumienia danych audio między samym kodekiem a procesorem realizowane jest poprzez dwukierunkowy port szeregowy McBSP0. Kodek pracuje w trybie master, co oznacza, że to on generuje sygnały zegarowe oraz ramki synchronizacyjne. DSP ma do dyspozycji 6 niezależnych kanałów DMA. Daje to możliwość transferu danych z pamięci TMS320VC5416 bez potrzeby angażowania CPU.

**Klawiatura**

Klawiatura podłączana jest do portu PS2. Układ 74LCX125 pełni rolę izolacji pomiędzy mikrokontrolerem zasilanym napięciem 3,3 V a poziomami o wartości 5 V pochodzącymi z klawiatury. W momencie, kiedy MSP430F135 odczyta kod naciśniętego klawisza za pomocą funkcji KeyboardRead(), zapisuje go w buforze transmisji SPI. Następnie sygnalizuje procesorowi DSP przerwaniem INT2 o gotowości do wysłania informacji o naciśniętym klawiszu.

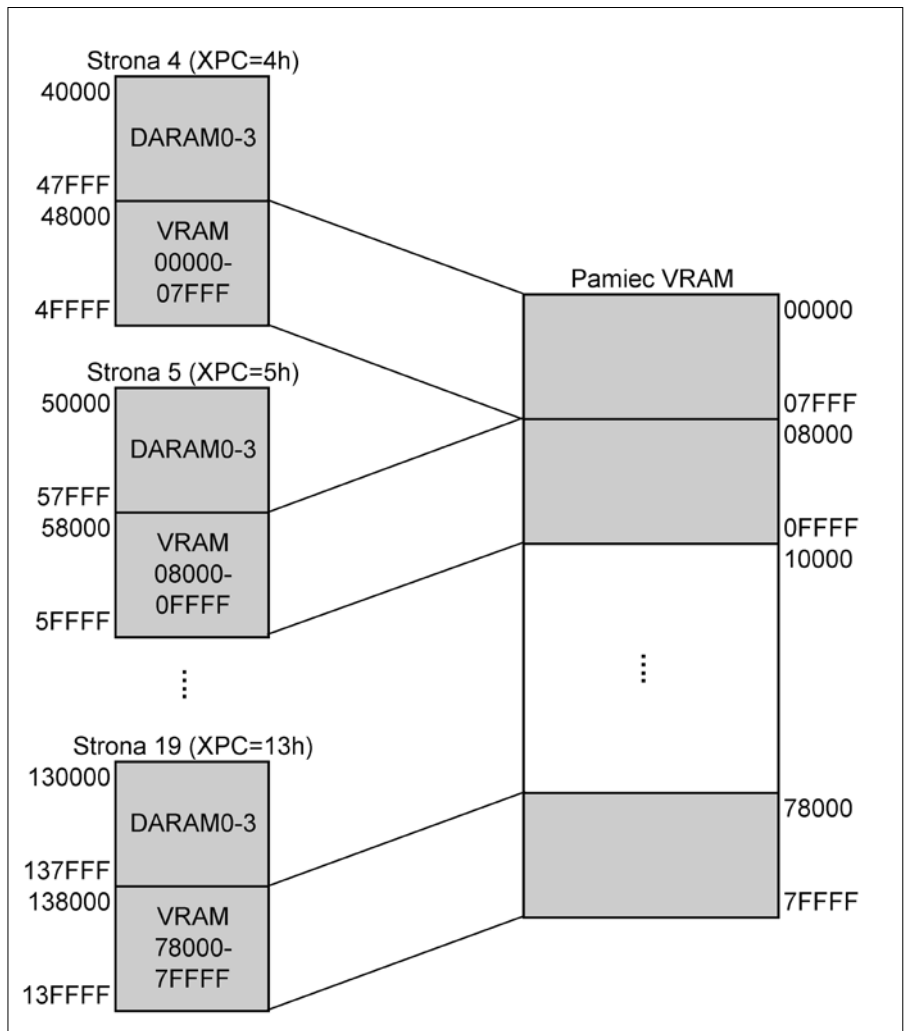
**Wyprowadzenia**

Mając na uwadze możliwość dalszej rozbudowy systemu, na płytce umieszczono dwa złącza (rys. 3). Pierwsze to wielokanałowy buforowany port szeregowy McBSP1. Drugie złącze wyprowadza równoległy interfejs HPI8, w którym adres i dane są multiplexowane na jednym 8-bitowym porcie. Możliwy jest przez niego dostęp do wewnętrznej pamięci TMS320VC5416. Obydwa interfejsy można skonfigurować jako GPIO. Dodatkowo zworka JP1 pozwala na aktywowanie HPI8 sprzętowo, tuż po zerowaniu procesora.

Poza samymi złączami do wymiany danych, na płytce znajdują się także porty JTAG dla XC95288XL, MSP430F135 i TMS320VC5416 pozwalające na programowanie w systemie.

**Zasilanie**

Głównym żywicielem zestawu jest przetwornica MAX1626. Jej maksymalna wydajność prądowa wynosi ponad 2 A. Układy w zestawie nie mają wprawdzie takiego apetytu, jednak trzeba uwzględnić możliwość podłączenia ekranu LCD. Zasilanie wyprowadzono na złączu CON6, gdzie oprócz napięcia 3,3 V bezpośrednio dostępne jest napięcie wejściowe układu. Procesor



Rys. 12. Organizacja pamięci wideo w przestrzeni adresowej DSP

TMS320VC5416 wymaga dodatkowo napięcia o wartości 1,6 V do zasilania rdzenia. Służy do tego stabilizator LK112M16. W celu uzyskania odpowiedniej jakości dźwięku część analogową kodeka TLV320AIC23 zasilają stabilizator LP3985.

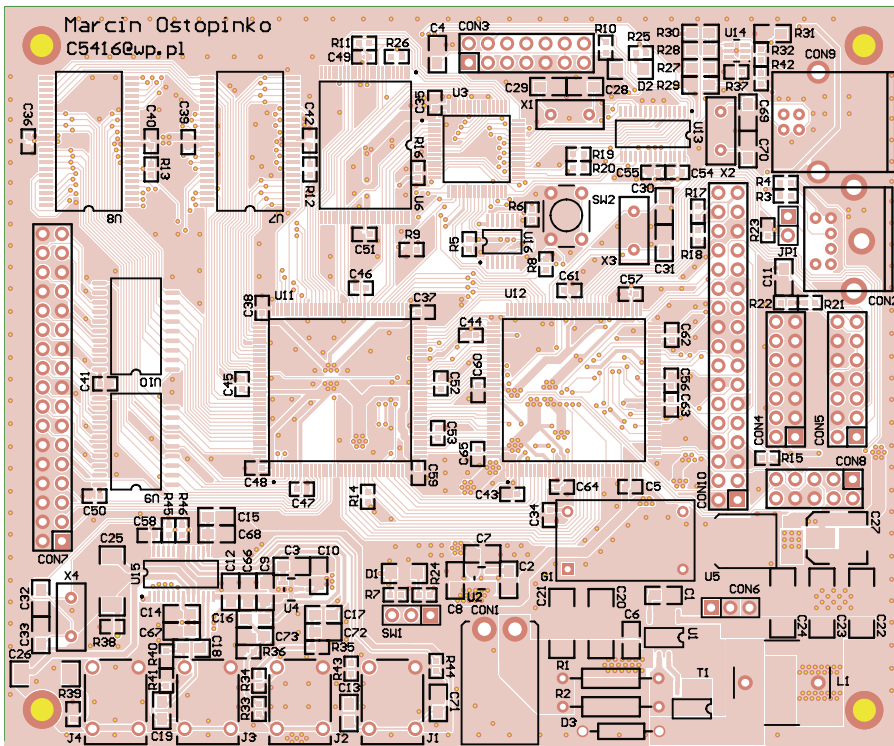
**Montaż i uruchomienie**

Schemat montażowy płytki drukowanej pokazano na rys. 13. Zmontowanie układu wymaga sporych umiejętności w lutowaniu

SMD. Układy TMS320VC5416 i XC95288XL mają dosyć drobny raster, a dodatkowo blisko nich umieszczone są kondensatory w obudowach 0603. W pierwszej kolejności lutujemy przetwornicę MAX1626 wraz z elementami potrzebnymi do jej pracy (R1, R2, C1, C6, C20...24, L1, D3, T1) oraz mikrokontroler MSP430F135. Poprzez JTAG-a na złączu CON3 zastępujemy stary program z Nokii nowym, właściwym dla naszego systemu. Można tego dokonać za

Tab. 8. Mapa portów w przestrzeni adresowej I/O		
Adres	Dostęp	Opis
0000h	RW	Interfejs USB – zapis/odczyt danych
0001h	RW	Interfejs USB – zapis/odczyt komend
0003h	RW	Flash – zapis(W=0)/odczyt(R=0) danych(CS=ALE=CLE=0), adresu (ALE=1 CS=CLE=0), komendy (CS=ALE=0 CLE=1)
0004h	W	Flash – zapis do tego portu ustawia CS=0
0005h	W	Flash – zapis do tego portu ustawia CS=1
0006h	W	Flash – zapis do tego portu ustawia ALE=1
0007h	W	Flash – zapis do tego portu ustawia ALE=0
0008h	W	Flash – zapis do tego portu ustawia CLE=1
0009h	W	Flash – zapis do tego portu ustawia CLE=0
000Ah	W	Flash – zapis do tego portu ustawia R=0
000Bh	W	Flash – zapis do tego portu ustawia R=1
000Ch	W	Flash – zapis do tego portu ustawia W=0
000Dh	W	Flash – zapis do tego portu ustawia W=1





Rys. 13. Schemat montażowy

pomocą JTAG-a opisanego w EP 3/2005 (AVT-1409). Programowanie układu w późniejszej fazie montażu może spowodować nieoczekiwane efekty, ponieważ w wyniku działania starego programu mogą wystąpić konflikty na

liniach I/O z innymi układami. Następnie można dokonać montażu reszty elementów na PCB poza procesorem TMS320VC5416. Przed wykonaniem tej czynności trzeba zaprogramować układ XC95288XL, także przez JTAG znajdujący

się na złączu CON4. Gdy już mamy poprawnie zmontowany układ, podłączamy klawiaturę i przesuwamy przełącznik SW1 w kierunku przetwornicy. Po podłączeniu zasilania (6...15 V) jednocześnie wciskamy klawisze Ctrl+Alt+Esc. W tym momencie jest „formatowana” pamięć NAND Flash, czyli kasowane są w niej dane i przygotowana jest odpowiednia struktura w celu obsługi systemu plików. Po odczekaniu kilku sekund podłączamy zestaw poprzez port USB do komputera. System Windows powinien wykryć układ jako dysk wymienny, na który kopiujemy program demonstracyjny (*boot.dat*). W celu uruchomienia programu przesuwamy przełącznik SW1 w kierunku kodeka audio. Na ekranie LCD jest widoczny system plików w pamięci NAND Flash. Nawigacja po nim odbywa się kursorami oraz klawiszami Enter i Esc. Możliwy jest podgląd plików graficznych BMP oraz odgrywanie dźwięku w formacie WAV. Program ten to oczywiście zaledwie ułamek możliwości zestawu. Przykładowo, podłączając przetwornik ADC można w prosty sposób zbudować oscyloskop. Natomiast poprzez zastosowanie odbiornika GPS możemy wyświetlać pozycję na ekranie LCD z użyciem map zapisanych w pamięci Flash. Przy wykorzystaniu mostka PCI-2040 podłączanego do interfejsu HPI8, zestaw może nawet współpracować z magistralą PCI.

**Marcin Ostopinko**  
C5416@wp.pl

R E K L A M M A

Sieci internetowe nie są zarezerwowane wyłącznie dla komputerów. W prostych aplikacjach doskonałym rozwiązaniem jest zastosowanie (zamiast PC) urządzenia opartego na mikrokontrolerze. Sprawdza się to doskonale w przypadku zdalnego sterowania lub pomiarów. Mikrokontroler jest znacznie mniejszy (od komputera), zużywa mniej prądu i jest mniej podatny na wszelkie „wpadki” programowo-obsługowe.

Aby mikrokontroler współpracował z Internetem potrzebny jest układ sprzęgający – **interfejs ethernetowy**. Oto moduł (AVT 1443) uniwersalnego interfejsu ethernetowego, który zawiera wszystkie elementy niezbędne do budowy toru Ethernet-procesor.

Możliwości zastosowań tego modułu do różnorodnych aplikacji, korzystających z Internetu, są praktycznie nieograniczone.

#### Zadanie konkursowe

Prześlij do redakcji EP (e-mail [redakcja@ep.com.pl](mailto:redakcja@ep.com.pl)) opis projektu (ze schematem) układu, w którym chcesz wykorzystać moduł AVT 1443. Wszystkim autorom oryginalnych, poprawnych propozycji wyślemy za darmo moduł AVT 1443. Jesteśmy też zainteresowani opublikowaniem wykonanych i sprawdzonych układów z zastosowaniem interfejsu Ethernet.