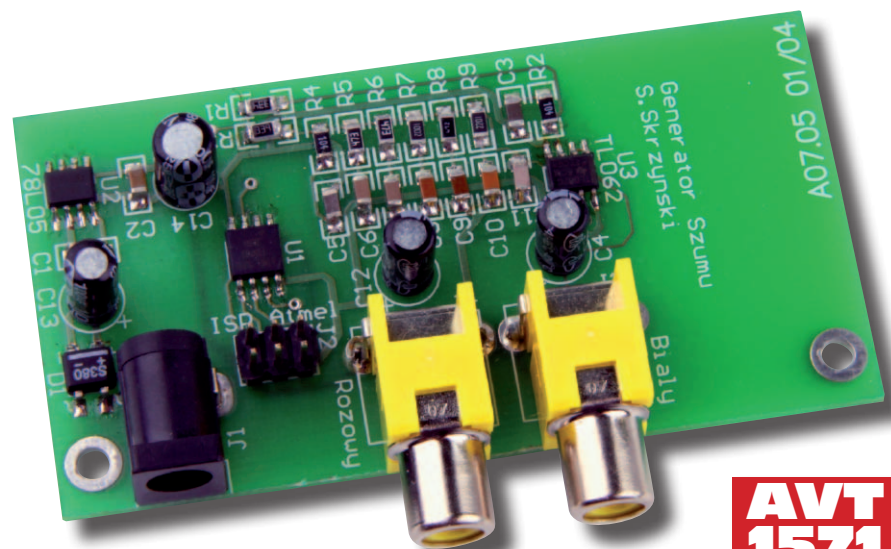


# Generator szumu różowego

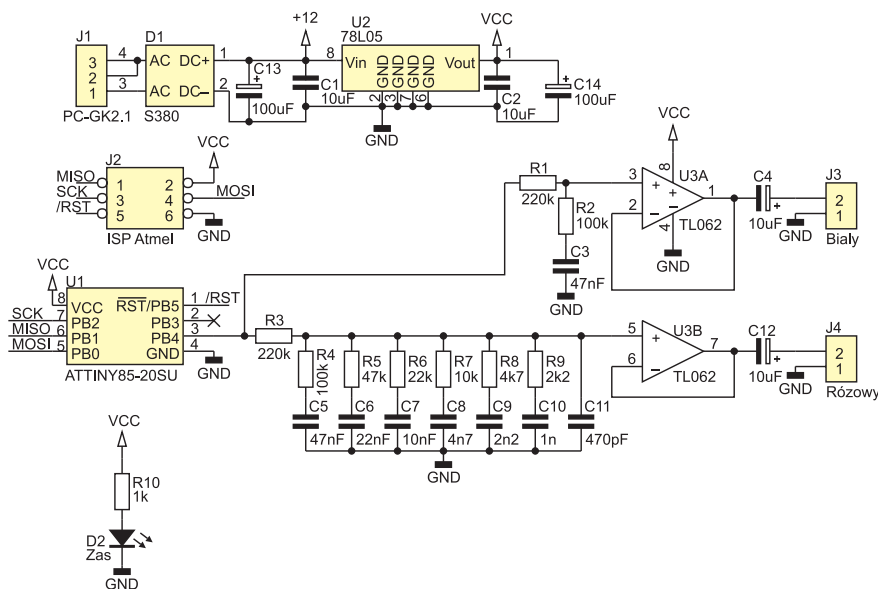
W elektroakustyce do ustawienia parametrów audio używa się generatora szumu. Podany na wejście urządzenia szum bada się na wyjściu analizatorem widma. Podobne badanie przy użyciu przestrajanego generatora i miernika poziomu sygnału trwa dość długo. Generator szumu i analizator dają wynik w ułamku sekundy. Dzięki temu na bieżąco możemy widzieć zmiany wnoszone przez np. korektor barwy dźwięku.

Badaniu pasma przenoszenia można poddać wzmacniacz, głośniki czy mikrofon. Szum może pomóc ustawić korektorem graficznym płaską charakterystykę przenoszenia zestawu audio, łącznie z głośnikami. Szum można wygenerować przy użyciu komputera z kartą dźwiękową. Rozwiązanie jest proste, ale może nie być zbyt wygodne, jeśli nie dysponujemy laptopem.

W Internecie można znaleźć wiele schematów generatorów. Najczęściej są to ukła-



dy oparte o szumiące złącze PN tranzystora lub diody. Generatory takie są proste, ale ich parametry nie są powtarzalne. Dobry generator można zbudować na procesorze sygnałowym. Niestety, takie procesory nie są zbyt tanie. Używając jednak generatora pseudolosowego zbudowanego z rejestru przesuwnego i bramki exor oraz filtru, można zbudować



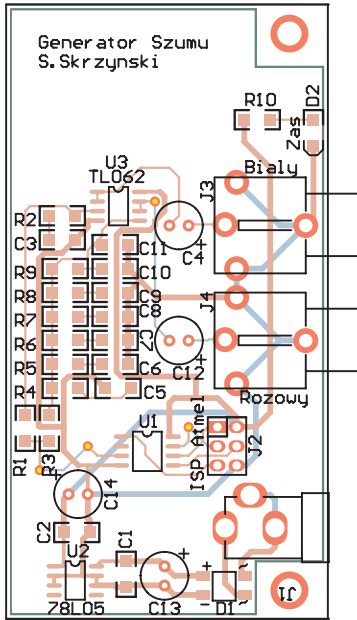
Rys. 1. Schemat ideowy generatora szumu

**AVT-1571 w ofercie AVT:**  
 AVT-1571A – płytka drukowana  
 AVT-1571A – płytka drukowana + elementy

**Dodatkowe materiały na CD i FTP:**  
<ftp://ep.com.pl>, user: 15257, pass: 1ajs046  
 • wzory płytek PCB  
 • karty katalogowe i noty aplikacyjne elementów oznaczonych na Wykazie Elementów kolorem czerwonym

- Wykaz elementów**
- R1, R3: 220 kΩ
  - R2, R4: 100 kΩ
  - R5: 47 kΩ
  - R6: 22 kΩ
  - R7: 10 kΩ
  - R8: 4,7 kΩ
  - R9: 2,2 kΩ
  - R10: 1 kΩ
  - C1, C2: 10uF (1206) C4, C12: 10uF/16 V
  - C3, C5: 47 nF
  - C6: 22 nF
  - C7: 10 nF
  - C8: 4,7 nF
  - C9: 2,2 nF
  - C10: 1 nF
  - C11: 470 pF C13, C14: 100uF/16V
  - U1: ATtiny85-20SU
  - U2: 78L05
  - U3: TL062
  - D1: S380 mostek prostowniczy
  - D2: dioda LED
  - J1: PC-GL2.1 złącze
  - J2: goldpin 3×2
  - J3, J4: goldpin 2×1





Rys. 2. Schemat montażowy generatora szumu

tani generator o zadowalających parametrach. Chciałem generator zbudować w taki właśnie sposób, ale przyszedł mi do głowy inny pomysł. Generator na rejestrach zastąpiłem mikrokontrolerem. Dzięki temu można eksperymentować z różnymi algorytmami pseudolosowymi.

Schemat generatora pokazano na rys. 1. Zastosowałem mikrokontroler AVR ATtiny85. Wybór procesora był podyktowany jego małymi wymiarami, niską ceną oraz bardzo niską ceną debuggera AVR Dragon. W docelowym rozwiązaniu wystarczy ATtiny25, ponieważ program zajmuje niecałe 700 b.

Sygnal pseudolosowy jest filtrowany w pasywnych filtrach RC (R4-R9 i C5-C11). Odfiltrowany sygnał jest buforowany przez wzmacniacz operacyjny U3B. Na wyjściu wzmacniacza dostępny jest sygnał szumu różowego (złącze J4). Dodatkowo, sygnał bez filtrowania trafia na bufor U3A. Dzielnik R1/R2/C3 ogranicza amplitudę sygnału na wyjściu U3A do poziomu zbliżonego do sygnału na wyjściu U3B. Ponadto, dzięki zastosowaniu w dzielniku kondensatora, wysokie częstotliwości leżące poza pasmem akustycznym są odcinane, dzięki czemu w szumie są ograniczone wyższe harmoniczne. Na wyjściu J3 bufora dostępny jest sygnał szumu białego.

### List. 1. Najważniejszy fragment programu

```
//Pętla generująca szum - LFSR
void LosLSFR()
{
  while(1)
  {
    ClkEor; // Do pomiaru częstotliwości zegarowej
    wdt_reset(); // Reset WDG
    if ( (_rand(255) & 1) == 0 ) // Zależnie od wylosowanej liczby
      OutH; // ustaw wyjście w stan wysoki
    else
      OutL; // lub niski
  }
}

//Generuje liczbę pseudolosową
int _rand(unsigned int zakres)
{
  static unsigned long lfsr = 1;
  lfsr = (lfsr >> 1) ^ (-(lfsr & 1) & 0xd0000001);
  return( lfsr );
}
```

Generator może być zasilany napięciem w zakresie 8...15 V, AC lub DC, ponieważ ma własny prostownik i stabilizator.

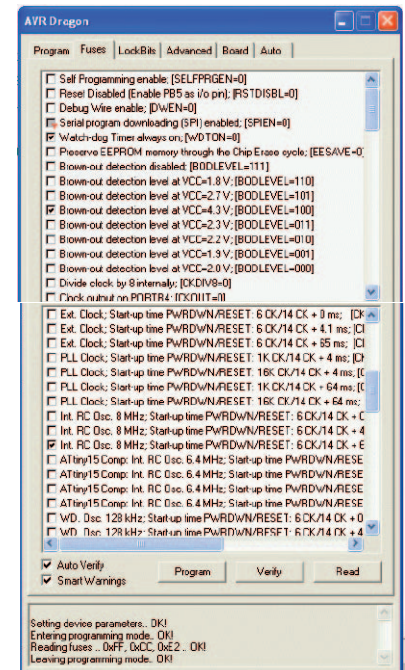
### Program

Dzięki zastosowaniu mikrokontrolera program jest banalny. Napisano go w języku C, w bezpłatnym środowisku AVR Studio. Do generowania liczb pseudolosowych C udostępniła funkcję rand(). Niestety funkcja ta wykonuje się dość długo, przez co umożliwia przy zegarze 8 MHz generowanie szumu z częstotliwością około 4,5 kHz. Próbowałem użyć innego algorytmu. Wykonywał się około 3 razy szybciej, ale to wciąż za mało, aby generować dobry szum. Zastosowałem więc metodę LFSR ([http://en.wikipedia.org/wiki/Linear\\_feedback\\_shift\\_register](http://en.wikipedia.org/wiki/Linear_feedback_shift_register)). Dzięki temu uzyskałem częstotliwość generowania szumu około 120 KHz, co przy ciągu 32-bitowym, daje powtórzenie po 35 sekundach. Najistotniejszy fragment kodu programu znajduje się na list. 1. Pierwszą czynnością po uruchomieniu programu jest ustawienie kierunku portów procesora. Następnie funkcja srand() inicjalizuje generator pseudolosowy.

W głównej pętli sprawdzany jest stan wejścia PB0. Jeśli wejście to jest w stanie wysokim, to szum jest generowany funkcją LFSR. Gdy wyprowadzenie to jest podłączone do masy, szum jest generowany funkcją rand(). Wyprowadzenie PB0 jest testowane tylko raz po restarcie procesora.

### Montaż i uruchomienie

Schemat montażowy generatora umieszczono na rys. 2. Montaż generatora przeprowadzamy w sposób klasyczny. Po wlutowaniu wszystkich elementów (poza mikrokontrolerem)



Rys. 3. Bity konfiguracji

rem) podłączamy zasilanie. Następnie kontrolujemy wartość napięcia na wyjściu stabilizatora. Jeśli napięcie jest poprawne, można zamontować mikrokontroler. Można go zaprogramować przed wlutowaniem lub po nim. Służy do tego 6-pinowe złącze J2. Ustawienie bitów konfiguracyjnych procesora przedstawiono na rys. 3.

Po poprawnym zaprogramowaniu procesora urządzenie podejmie pracę. Sygnal szumu można skontrolować, podłączając do wyjścia wzmacniacz lub słuchawki.

Płytką jest przeznaczona do umieszczenia w obudowie KM35.

Sławomir Skrzyński, EP  
slawomir.skrzynski@ep.com.pl

www.sklep.avt.pl