

# Spectrum – prosty analizator widma sygnału akustycznego

Są takie projekty, które swoją funkcjonalnością zabierają nas w lata 80-te, gdzie królowały polskie układy scalone spod znaku CEMI z dumą stosowane przez elektroników. Tych elektroników, co warto podkreślić, którym było dane kupić ten czy inny układ rodzimej produkcji co wcale nie było łatwe. Dość dobrze pamiętam te chwile, gdy zdobycie kultowego układu scalonego wskaźnikaysterowania typu UL1980 czy UL1970 granoczyło niemalże z cudem, zaś zbudowanie docelowego urządzenia było niemałym wyzwaniem. Sam niejednokrotnie przemierzałem niezapomnianym PKS-em dziesiątki kilometrów w poszukiwaniu sklepu elektronicznego, który dysponował interesującymi mnie układami. Dzisiaj, w czasach wszechobecnych mikrokontrolerów, zadanie takie możemy zrealizować software'owo rozbudowując przy okazji funkcjonalność urządzenia docelowego.

Na bazie takich założeń powstał projekt prostego analizatora widma sygnału akustycznego pod postacią projektu Spectrum, który w swojej implementacji łączy zalety sprzętowej i programowej realizacji oczekiwanej funkcjonalności. Sprzętowej, gdyż w budowie urządzenia wykorzystano bardzo ciekawy układ analizatora widma pod postacią układu MSGEQ7, który integruje w sobie 7 filtrów pasmowo-przepustowych o częstotliwościach środkowych 63 Hz, 160 Hz, 400 Hz, 1 kHz, 2,5 kHz, 6,25 kHz oraz 16 kHz, 7 detektorów wartości szczytowej oraz analogowy multiplekser wyjściowy. Schemat funkcjonalny układu MSGEQ7 pokazano na **rysunku 1**. Wejściowy sygnał audio podawany jest na wejścia 7 filtrów pasmowo-przepustowych, których wyjścia podłączono do detektorów wartości szczytowej. Wyjścia

detektorów podłączone są z kolei do 7-kanałowego multipleksa analogowego, którego wyjście wyprowadzono na zewnątrz układu MSGEQ7. Dzięki takiej budowie na wyjściu układu MSGEQ7 (OUT) otrzymujemy napięcie stałe odpowiadające wartości szczytowej dla wybranego filtra pasmowo-przepustowego, którego to wyboru dokonujemy dzięki cyfrowym wejściom STROBE i RESET. Przebiegi sygnałów sterujących na wejściach STROBE i RESET odpowiedzialnych za wystawianie kolejnych wartości szczytowych wbudowanych filtrów pasmowo-przepustowych pokazano na **rysunku 2**.

Stan wysoki na wejściu RESET zeruje wewnętrzny multiplekser analogowy i jednocześnie zatrzymuje wartości szczytowe dla wszystkich pasm częstotliwościowych.

Dodatkowe materiały do pobrania ze strony [www.media.avt.pl](http://www.media.avt.pl)

**W ofercie AVT\* AVT-5712**

#### Podstawowe parametry:

- liczba wyświetlanych pasm częstotliwości: 7
- częstotliwości środkowe pasm: 63 Hz, 160 Hz, 400 Hz, 1 kHz, 2,5 kHz, 6,25 kHz oraz 16 kHz;
- liczba sposobów prezentacji widma sygnału: 5
- maksymalna amplituda analogowego sygnału wejściowego: 5 V
- zasilanie: 5 VDC, 200 mA

#### Projekty pokrewne na [www.media.avt.pl](http://www.media.avt.pl):

- AVT-5678 Stereofoniczny wskaźnikysterowania (EP 6/2019)
- AVT-5585 Sterownik wskaźnika wychyłowego do wzmacniacza (EP 1/2018)
- AVT-1716 Wskaźnikysterowania z pamięcią (EP 12/2012)
- AVT-1517 Wskaźnik nie tylkoysterowania (EP 9/2012)
- AVT-5219 Wizualizator do Winampa na USB (EP 1/2010)
- AVT-5210 Analizator widma sygnału audio (EP 11/2009)
- AVT-2864 Analogowo-cyfrowy analizator widma (Edw 5/2008)
- AVT-580 Procesor audio z equalizerem i analizatorem widma (EP 6-7/2004)
- AVT-2375 Wskaźnikysterowania 2x5 LED (Edw 9/1999)
- AVT-2353 Pseudoanalogowy VU-metr (Edw 4/1999)

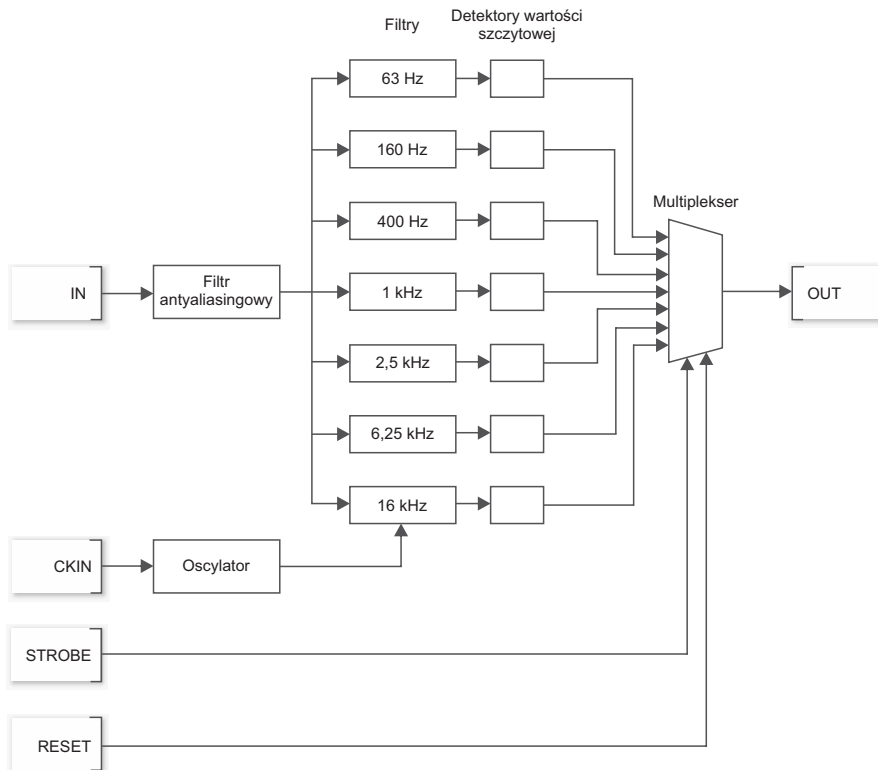
#### Uwaga! Elektroniczne zestawy do samodzielnego montażu.

wymagają umiejętności lutowniczych!

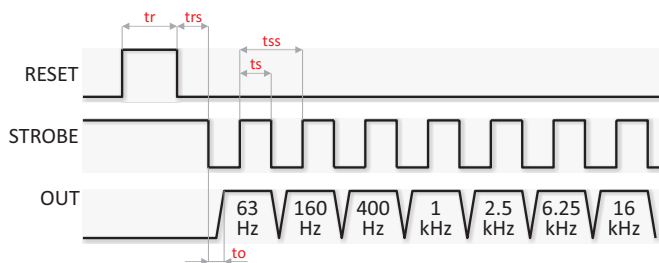
Podstawowy zestaw jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wzlutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu.

Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wzlutowane w płytkę PCB)
  - wersja [A] – płytkę drukowaną bez elementów i dokumentacji
  - wersja [A+] – płytkę drukowaną [A] + zaprogramowany układ [UK] i dokumentację
  - wersja [UK] – zaprogramowany układ
- Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz!
- <http://sklep.avt.pl>. W przypadku braku dostępności na <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: [kity@avt.pl](mailto:kity@avt.pl).



Rysunek 1. Schemat funkcjonalny układu MSGEQ7



tr - reset pulse: minimum 100ns  
 trs - reset to strobe: minimum 72us  
 ts - strobe pulse: minimum 18us  
 tss - strobe to strobe: minimum 72us  
 to - strobe settling time: minimum 36us

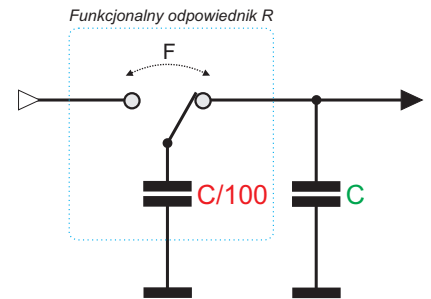
Rysunek 2. Przebiegi sygnałów sterujących na wejściach STROBE i RESET układu MSGEQ7

Następnie, każde opadające zbocze sygnału na wejściu STROBE powoduje wystawienie na wyjściu układu OUT kolejnej wartości szczytowej. Uważny czytelnik zauważy zapewne brak jakichkolwiek elementów pasywnych w torze przetwarzania sygnału, które sugerowałyby realizację filtrów pasmowo-przepustowych. Racja! W naszym układzie analizatora widma MSGEQ7 zastosowano filtry cyfrowe, których częstotliwości środkowe zależą w pewnym stopniu od elementów oscylatora napędzającego układ scalony tj. elementów R1/C2 podłączonych do wejścia sygnału taktującego CKIN. Filtry, o których mowa powyżej to tzw. filtry z przełączaną pojemnością.

Przybliżmy zatem to ciekawe i mało znane rozwiązanie. W filtrach z przełączaną pojemnością typowy obwód RC zastąpiony jest przez układ z dwoma kondensatorami i przełącznikiem, przy czym dodany kondensator ma znacznie mniejszą pojemność niż kondensator z standardowego obwodu RC (około 1%)

i razem z przełącznikiem stanowi niejako element zastępczy dla rezystora (patrz rysunek 3). W układzie takim, scalony przełącznik łączy ten mały kondensator naprzemiennie do wejścia i wyjścia z bardzo dużą częstotliwością sięgającą 100-krotności częstotliwości odcięcia filtru. Z uwagi na małą pojemność dołączonego kondensatora zostaje on dość szybko naładowany do chwilowej wartości napięcia wejściowego zaś w położeniu drugim (dołączony do wyjścia) przekazuje on swój ładunek do kondensatora o większej pojemności. W typowym układzie RC szybkość ładowania kondensatora zależy od wartości rezystora zaś w układzie filtrów z przełączaną pojemnością, zależy od pojemności dodatkowego kondensatora i częstotliwości przełączania, co czyni je bardzo elastycznymi, jeśli chodzi o zmianę częstotliwości odcięcia filtru, która to w układzie modelowym jest wprost proporcjonalna do częstotliwości taktowania zastosowanej do sterowania przełącznikiem. Tyle

**Ustawienia Fuse-bitów (ważniejszych):**  
 CKSEL3...0: 0010  
 SUT1...0: 10  
 CKDIV8: 0  
 EESAVE: 0



Rysunek 3. Uproszczony schemat funkcjonalny cyfrowego filtra aktywnego z przełączaną pojemnością

**Wykaz elementów:**

**Rezystory:** (obudowy SMD 0805)

R1: 200 kΩ  
 R2, R10: 22 kΩ  
 R3...R9: 2,2 kΩ  
 R11, R12, R18: 470 Ω  
 R13...R17, R19, R20: 75 Ω

**Kondensatory:** (obudowy SMD 0805)

C1, C3, C5...C7: 100 nF  
 C2: 33 pF  
 C4: 10 nF

**Półprzewodniki:**

U1: MSGEQ7 (DIL-08)  
 U2: ATmega48 (DIL-28)  
 T1...T7: BC847 (SOT23)  
 LED1...LED7: linijka diodowa zielono-czerwona typu OSX10201-GGR1 lub podobna

**Inne:**

L1: dławik 10 μH (SMD 0805)  
 MODE: mikroswitch TACT 9 mm

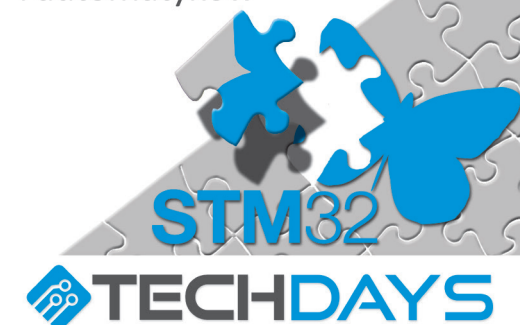
w kwestii naszego arcydzieła peryferium, przejdźmy zatem do schematu urządzenia Spectrum, który to pokazano na **rysunku 4**.

**Budowa**

Zaprojektowano bardzo prosty układ mikroprocesorowy, którego sercem jest niewielki mikrokontroler ATmega48 taktowany wewnętrznym, wysokostabilnym sygnałem zegarowym o częstotliwości 1 MHz. Mikrokontroler ten odpowiada za realizację całej, założonej funkcjonalności urządzenia tj.: zarządza pracą scalonego analizatora widma MSGEQ7 korzystając z jego wejść sterujących RESET i STROBE,

REKLAMA

Specjalistyczne szkolenia dla elektroników i automatyków



techdays@techdays.pl  
 TECHDAYS.PL

CERTYFIKOWANY PARTNER SZKOLENIOWY

Listing 1. Plik nagłówkowy do obsługi układu MSGEQ7

```
#define MSGEQ7_PORT PORTC
#define MSGEQ7_DDR DDRC
#define MSGEQ7_STROBE_NR PC2
#define MSGEQ7_RESET_NR PC4

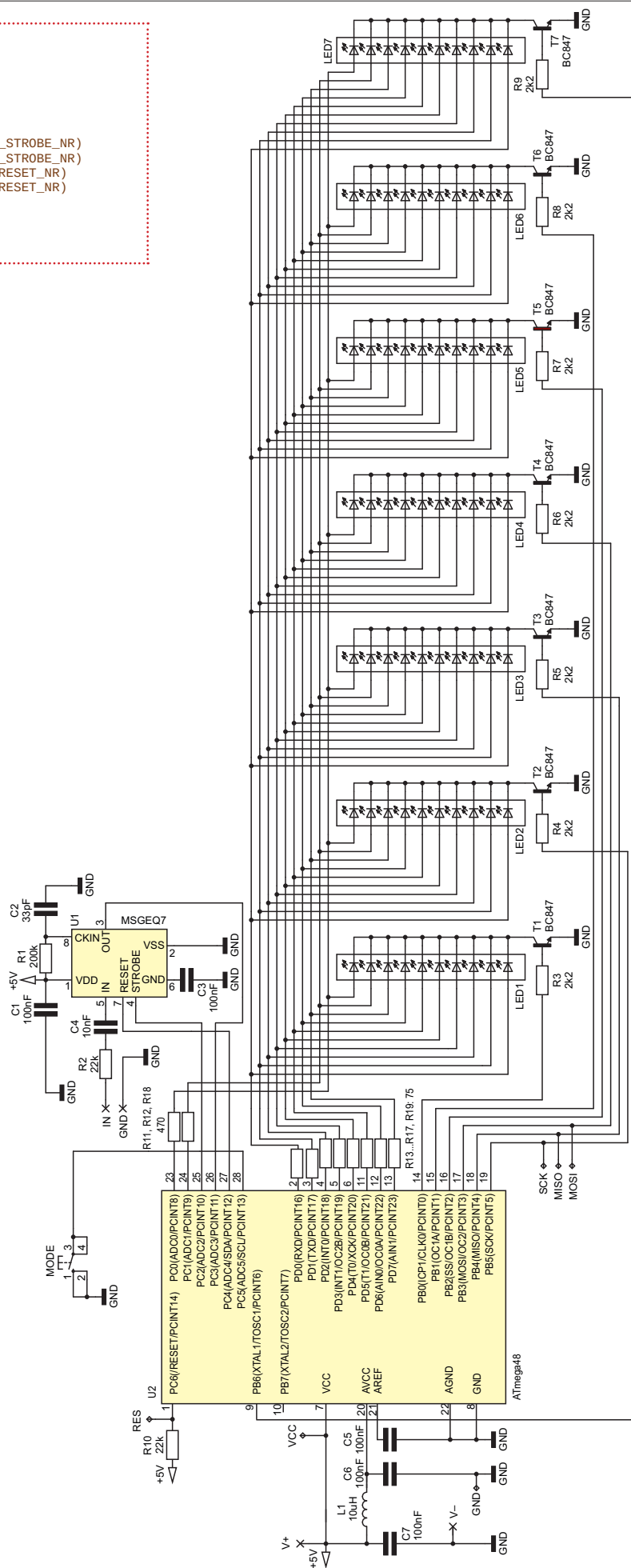
#define MSGEQ7_STROBE_HIGH MSGEQ7_PORT |= (1<<MSGEQ7_STROBE_NR)
#define MSGEQ7_STROBE_LOW MSGEQ7_PORT &= ~(1<<MSGEQ7_STROBE_NR)
#define MSGEQ7_RESET_HIGH MSGEQ7_PORT |= (1<<MSGEQ7_RESET_NR)
#define MSGEQ7_RESET_LOW MSGEQ7_PORT &= ~(1<<MSGEQ7_RESET_NR)

//Prototypy funkcji
void MSGEQ7init(void);
void MSGEQ7readBands(uint8_t *Bands);
```

mierzy napięcie na wyjściu tegoż układu (OUT) korzystając z wbudowanego w swoją strukturę przetwornika analogowo-cyfrowego (kanał ADC3), steruje pracą wyświetlacza LED zbudowanego z siedmiu 10-punktowych linii LED połączonych w konfiguracji wspólnej katody oraz realizuje obsługę przycisku MODE przeznaczanego do zmiany sposobu prezentacji widma sygnału.

Sterowanie pracą wyświetlacza LED zrealizowane zostało z wykorzystaniem mechanizmu multipleksowania, za który odpowiedzialny jest wbudowany w mikrokontroler układ czasowo-licznikowy Timer0 skonfigurowany w taki sposób by generował przerwanie od przepełnienia (tryb CTC układu) 420 razy na sekundę, czyli 60 razy na każdą z linii LED. W przerwaniu takim w pierwszej kolejności wygaszane są wszystkie wspólne katody wyświetlaczy LED, następnie na porty PD0...PD7 oraz PC0 i PC1 wystawiana jest wartość odpowiadająca zmierzonemu napięciu na wyjściu OUT układu MSGEQ7, po czym włączana jest kolejna wspólna katoda sterowana z portów PB0...PB6 mikrokontrolera (z wykorzystaniem tranzystorów sterujących T1...T7). Jest to typowe rozwiązanie mechanizmu multipleksowania stosowane w wielu systemach mikroprocesorowych, dzięki któremu możliwe jest sterowanie dużą liczbą diod LED z wykorzystaniem ograniczonej liczby portów mikrokontrolera. Dzięki temu ograniczamy przy okazji sumaryczny prąd pobierany przez wyświetlacz LED, który w tym przypadku wynosi maksymalnie 10×10 mA. Tyle w kwestiach sprzętowych, w związku z czym przejdźmy do najważniejszych zagadnień programowych.

Na początek plik nagłówkowy do obsługi układu MSGEQ7, którego treść pokazano na **listingu 1**. Dalej, na **listingu 2**, dwie bardzo proste funkcje przeznaczone do obsługi układu MSGEQ7: inicjalizacyjna i pomiarowa. Jak widać, funkcja pomiarowa, której zadaniem jest pozyskanie wartości napięcia dla każdego z pasm częstotliwościowych, korzysta z prostej funkcji narzędziowej readADC(), której zawartość pokazano na **listingu 3**. Na koniec przedstawię podstawy implementacji mechanizmu multipleksowania. Zaczniemy od pliku nagłówkowego, którego zawartość pokazano na **listingu 4**. Dalej, na **listingu 5** pokazano funkcję inicjalizacyjną mechanizmu multipleksowania, która to ustawia odpowiednie stany logiczne na portach wspólnych katod i anod wyświetlacza LED oraz inicjuje stosowny timer sprzętowy. Kolejną,



Rysunek 4. Schemat ideowy analizatora Spectrum



prezentowaną funkcją jest funkcja obsługi przetwarzania realizująca mechanizm multipleksowania, której to zawartość pokazano na **listingu 6**. Wspomniana funkcja korzysta z deklaracji zmiennej umieszczonej w pamięci Flash mikrokontrolera, jak niżej:

```
const uint8_t commCathodes[7]
PROGMEM = {(1<<COM_CAT_NR1),
(1<<COM_CAT_NR2), (1<<COM_CAT_
NR3), (1<<COM_CAT_NR4),
(1<<COM_CAT_NR5), (1<<COM_CAT_
NR6), (1<<COM_CAT_NR7)};
```

**Listing 2. Funkcje przeznaczone do obsługi układu MSGEQ7**

```
void MSGEQ7init(void)
{
    //Ustawienie portów STROBE i RESET, jako wyjściowe ze stanem „0”
    MSGEQ7_DDR |= (1<<MSGEQ7_STROBE_NR)|(1<<MSGEQ7_RESET_NR);
    //Stan wyjściowy: Reset=0, Strobe=1
    MSGEQ7_RESET_LOW;
    MSGEQ7_STROBE_HIGH;
}

void MSGEQ7readBands(uint8_t *Bands) //920us
{
    MSGEQ7_RESET_HIGH;
    _delay_us(1); //Reset pulse width: : minimum 100ns
    MSGEQ7_RESET_LOW;
    _delay_us(75); //Reset to strobe time: minimum 72us

    for (uint8_t Nr = 0; Nr < 7; Nr++)
    {
        MSGEQ7_STROBE_LOW;
        _delay_us(40); //Output settling time: minimum 36us
        *Bands++ = readADC(); //60 us
        MSGEQ7_STROBE_HIGH;
        _delay_us(20); //Strobe to strobe: minimum 72us
    }
}
```

**Listing 3. Funkcja narzędziowa odpowiedzialna za pomiar z użyciem przetwornika ADC**

```
uint8_t readADC(void)
{
    //Wybór Vref=AVCC (5V), wejścia ADC3 przetwornika oraz
    //justowanie wyniku do lewej (dokładność 8 bitów)
    ADMUX = (1<<REFS0)|(1<<ADLAR)|0x03;
    //Uruchomienie konwersji, Preskaler=4 (250kHz @ 1MHz)
    ADCSRA = (1<<ADEN)|(1<<ADSC)|(1<<ADPS1);
    //Czekamy na zakończenie bieżącej konwersji - 60us
    while(ADCSRA & (1<<ADSC));

    return ADCH/24; //Zakres 0...10
}
```

**Listing 5. Funkcja inicjalizacyjna mechanizmu multipleksowania**

```
void initMultiplex(void)
{
    //Porty wspólnych anod, jako porty wyjściowe ze stanem nieaktywnym „0”
    COM_ANODE_LOWER_DDR = 0xFF;
    COM_ANODE_UPPER_DDR |= (1<<COM_ANODE_UPPER_LOW) | (1<<COM_ANODE_UPPER_HIGH);
    //Port wspólnych katod, jako port wyjściowy ze stanem nieaktywnym „0”
    COM_CAT_DDR = 0xFF;

    //Konfiguracja licznika Timer0 w celu generowania przerwania do
    //obsługi multipleksowania wyświetlacza LED (420 Hz)
    TCCR0A = (1<<WGM01); //Tryb CTC
    TCCR0B = (1<<CS02); //Preskaler = 256 @ 1MHz
    //420 Hz (przerwanie 420 razy na sekundę, 60 razy na sekundę)
    //dla każdego wyświetlacza LED)
    OCR0A = 8;
    //Uruchomienie przerwania Output Compare Match A (od porównania)
    TIMSK0 = (1<<OCIE0A);
}
```

**Listing 6. Funkcja obsługi przerwania realizująca mechanizm multipleksowania**

```
//Zmienna przechowująca wartość wyświetlaną na kolejnych słupkach wyświetlacza
uint16_t Led[7];

ISR(TIMERO0_COMPA_vect)
{
    static uint8_t Nr; //Numer kolejnego słupka do wyświetlenia
    uint16_t Value = Led[Nr]; //Optymalizacja dostępu

    //Wygaszenie wspólnych katod
    COM_CAT_BLANK;

    //Wystawienie właściwych stanów na portach wspólnych anod
    COM_ANODE_LOWER_PORT = Value & 0xFF;
    if(Value & 0b100000000) COM_ANODE_UPPER_PORT |= (1<<COM_ANODE_UPPER_LOW);
    else COM_ANODE_UPPER_PORT &= ~(1<<COM_ANODE_UPPER_LOW);
    if(Value & 0b1000000000) COM_ANODE_UPPER_PORT |= (1<<COM_ANODE_UPPER_HIGH);
    else COM_ANODE_UPPER_PORT &= ~(1<<COM_ANODE_UPPER_HIGH);

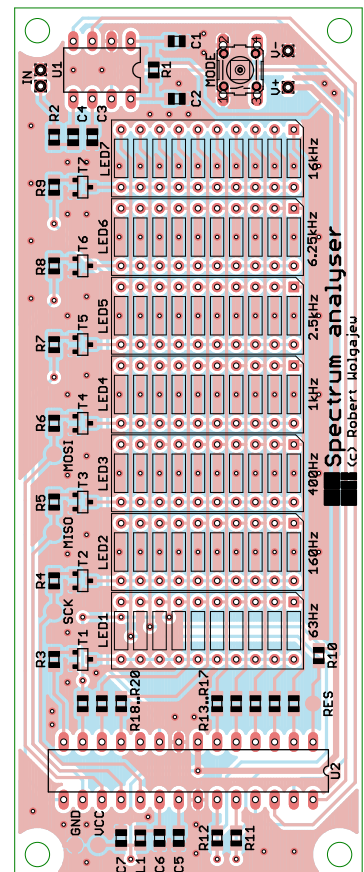
    //Załączenie odpowiedniej wspólnej katody
    COM_CAT_PORT = pgm_read_byte(&commCathodes[Nr]);
    //Wybranie kolejnego słupka LED
    if(++Nr > 6) Nr = 0;
}
```

**Listing 4. Plik nagłówkowy mechanizmu multipleksowania**

```
#define COM_ANODE_LOWER_PORT PORTD
#define COM_ANODE_LOWER_DDR DDRD
#define COM_ANODE_UPPER_PORT PORTC
#define COM_ANODE_UPPER_DDR DDRC
#define COM_ANODE_UPPER_LOW PC1
#define COM_ANODE_UPPER_HIGH PC0

#define COM_CAT_PORT PORTB
#define COM_CAT_DDR DDRB
#define COM_CAT_NR1 PB0
#define COM_CAT_NR2 PB5
#define COM_CAT_NR3 PB4
#define COM_CAT_NR4 PB3
#define COM_CAT_NR5 PB2
#define COM_CAT_NR6 PB1
#define COM_CAT_NR7 PB6
#define COM_CAT_BLANK COM_CAT_PORT = 0x00

//Zmienna przechowująca wartość
//wyświetlaną na kolejnych słupkach
//wyświetlacza
extern uint16_t Led[7];
```



**Rysunek 5. Schemat montażowy analizatora Spectrum**

REKLAMA

Specjalistyczne szkolenia dla elektroników i automatyków

STM32

STM32

TECHDAYS

techdays@techdays.pl  
TECHDAYS.PL

CERTYFIKOWANY PARTNER SZKOLENIOWY

ST

little augmented

**Listing 7.** Fragment programu obsługi aplikacji odpowiedzialny za wyświetlanie poszczególnych pasm częstotliwości dla każdego z trybów pracy analizatora

```
MSGEQ7readBands(Band); //920us
//Wartości poszczególnych pasm przeliczamy na kolejne słupki LED, przy czym sposób przeliczenia
//zależy od bieżącego trybu pracy urządzenia: wyświetlanie pojedynczych pasków czy słupków
for(uint8_t i = 0; i < 7; ++i)
{
    switch(workMode)
    {
        case MODE_POINT:
            //Rysujemy pojedynczą linię dla każdego wyświetlacza LED
            if(Band[i] == 0) Led[i] = 0; else Led[i] = (1<<(Band[i]-1));
            break;

        case MODE_BAR:
            //Rysujemy słupki dla każdego wyświetlacza LED
            Led[i] = 0;
            for(uint8_t Val = 1; Val<11; ++Val) if(Band[i] >= Val) Led[i] |= (1<<(Val-1));
            break;

        case MODE_BAR_MAX:
            //Aktualizujemy maksima
            if(Band[i] > maxBand[i]) maxBand[i] = Band[i];
            //Co 200 ms zmniejszamy maksima o jeden pasek w dół
            if(++timer200ms%4 == 0) if(maxBand[i] > 0) --maxBand[i];
            //Rysujemy słupki dla każdego wyświetlacza LED plus pasek maximum
            Led[i] = 0;
            for(uint8_t Val = 1; Val<11; ++Val) if(Band[i] >= Val) Led[i] |= (1<<(Val-1));
            //Dorysowanie paska maximum
            Led[i] |= (1<<(maxBand[i]-1));
            break;

        case MODE_BAR_NEGATIVE:
            //Rysujemy słupki dla każdego wyświetlacza LED
            Led[i] = 0x3FF;
            for(uint8_t Val = 1; Val<11; ++Val) if(Band[i] >= Val) Led[i] &= ~(1<<(Val-1));
            break;

        case MODE_BASELINE:
            Led[i] = 0;
            if(Band[i] >= 6){
                for(uint8_t Val = 6; Val<11; ++Val) if(Band[i] >= Val) Led[i] |= (1<<(Val-1));
            }else{ //5...1
                for(uint8_t Val = 1; Val<6; ++Val) if(Band[i] >= Val) Led[i] |= (1<<(5-Val));
            }
            break;
    }
}
```

Na koniec, dla dociekliwych Czytelników, fragment programu obsługi aplikacji (**listing 7**) odpowiedzialny za wyświetlanie poszczególnych pasm częstotliwości dla każdego z trybów pracy analizatora.

## Montaż i uruchomienie

Schemat montażowy urządzenia Spectrum, pokazano na **rysunku 5**. Zaprojektowano bardzo zgrabny i zwarty obwód drukowany ze zdecydowaną przewagą elementów SMD. Montaż urządzenia rozpoczynamy od przyłutowania tranzystorów sterujących T1...T7, następnie lutujemy rezystory, kondensatory i dławik SMD, a na samym końcu układy scalone, wyświetlacze LED oraz przycisk MODE. Poprawnie zmontowany układ nie wymaga żadnych regulacji i powinien działać tuż po włączeniu zasilania. Warto wspomnieć, iż na płytce urządzenia przewidziano punkty lutownicze do podłączenia zasilania (V+, V-), sygnału wejściowego (IN), jak i te, przy pomocy których zaprogramujemy nasz mikrokontroler (VCC, GND, RES, SCK, MISO, MOSI). Dla dociekliwych warto wspomnieć, iż program obsługi aplikacji urządzenia Spectrum przewiduje 5 trybów wyświetlania informacji o widmie sygnału audio, których przykładowe wizualizacje pokazano na **rysunku 6**. Warto podkreślić, że tryb trzeci (Bar maximum) integruje dodatkową funkcjonalność w postaci pokazywania wartości szczytowej w każdym z pasm częstotliwościowych.

Robert Wołgajew, EP



**Rysunek 6.** Wizualizacje przedstawiające 5 trybów wyświetlania informacji o widmie sygnału audio