



SAR – amplituner stereofoniczny (2)

Projekt amplitunera „chodził” za mną od dłuższego czasu, ale zawsze było coś ważniejszego do zrobienia, więc musiał „nabrać mocy urzędowej”. Nie bez znaczenia był w tym wypadku fakt, iż z założenia chciałem przygotować rozwiązanie kompleksowe, integrujące zarówno zagadnienia sprzętowe oraz projekt obudowy, a to wymaga znacznie więcej czasu i wysiłku. W drugiej części artykułu zajmiemy się opisem symulacji części analogowej tunera oraz opisem sposobu funkcjonowania jego oprogramowania.

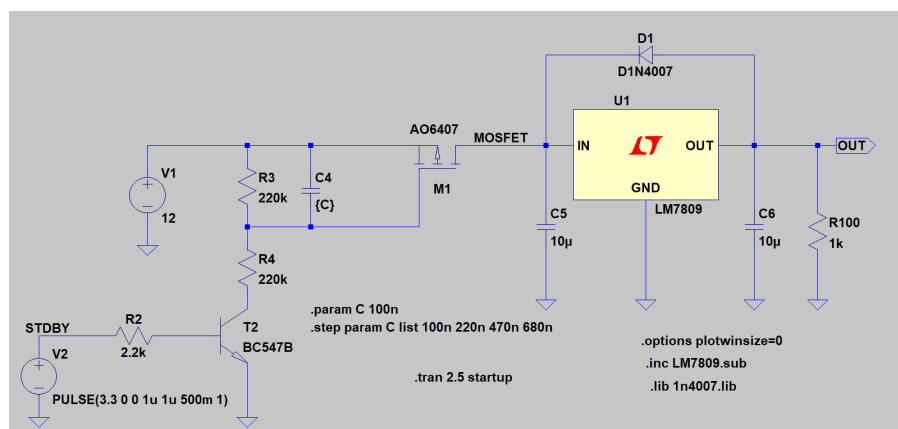
Symulator LTSpice jest bardzo dokładny, ale otrzymane wyniki zależą od poprawności użytych modeli. A z kolei poprawność modeli jest kompromisem między ich dokładnością, a czasem wykonania symulacji. Na pewno nikt nie chciałby czekać kilka godzin na wyniki symulacji. Zwłaszcza, że symulację należy powtarzać dla różnych wartości elementów. A zatem, wyniki symulacji są dobrą wskazówką odnośnie do zachowania się badanego urządzenia, ale dopiero działanie

rzeczywistego układu pozwoli zweryfikować ostatecznie jego funkcjonowanie. Niemniej jednak, sprawdzenie w symulatorze zdecydowanie przyspiesza prototypowanie.

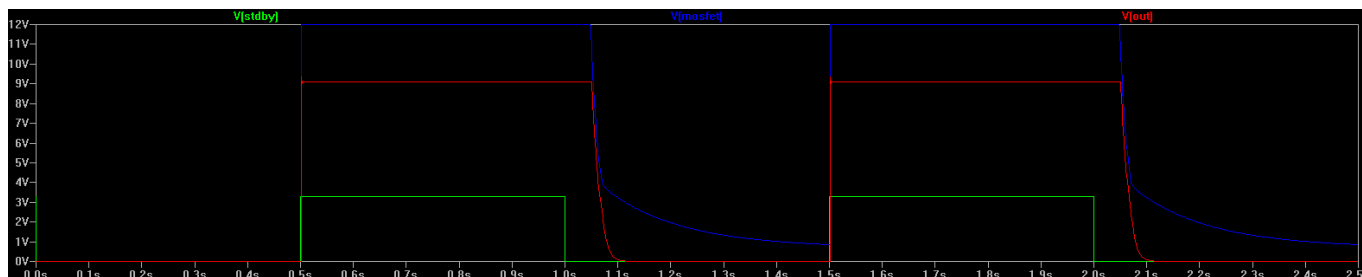
Zacznijmy od czegoś względnie nieskomplikowanego – od zasilacza niskiego napięcia LV POWER SUPPLY. Zasilacz jest sterowany sygnałem STB z procesora i ma pozostać wyłączony, jeśli urządzenie jest w trybie STANDBY i włączony po uruchomieniu urządzenia. Spójrzmy na symulację

pokazaną na **rysunku 6**. Sygnał sterujący jest zasymulowany jako źródło napięcia pracujące w trybie PULSE z napięciem o poziomie 3,3 V przełączanym co pół sekundy na wartość 0 V. Przy okazji możemy sprawdzić pojemność kondensatora C4, który decyduje po jakim czasie od wyłączenia układu przez procesor faktycznie zanika napięcie 9 V. Z tego powodu, jego pojemność jest zdefiniowana jako parametr. Następną instrukcja „step param C list 100n 220n 470n 680n” określa jakie wartości kondensatora chcemy przetestować, jednak tymczasem jest ona traktowana jako komentarz (kolor polecenia jest niebieski). Otrzymujemy wynik symulacji pokazany na **rysunku 7**. Sygnał V(stdby) o kolorze zielonym oznacza napięcie sterujące z procesora, sygnał V(mosfet) w kolorze niebieskim oznacza napięcie załączane przez MOSFET, a sygnał V(out) o kolorze czerwonym oznacza wyjściowe napięcie zasilacza. Napięcie wyjściowe jest prawidłowo sterowane przez napięcie dostarczane z procesora. Pomiedzy wyłączeniem napięcia procesora a wyłączeniem napięcia wyjściowego istnieje nieznaczne opóźnienie określone przez pojemność kondensatora C4. Dla pojemności 100 nF to opóźnienie wynosi około 50 ms, co można odczytać bezpośrednio z osi czasu lub po przypisaniu do napięcia V(out) dwóch kursorów dostępnych w LTSpice, jak pokazano na **rysunku 8**.

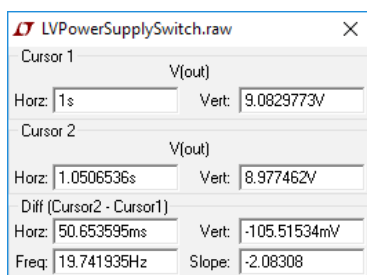
Włączmy teraz dyrektywę określającą listę wartości kondensatorów, które chcemy przetestować – **rysunek 9**. Po powtórnym uruchomieniu symulacji otrzymujemy następujące



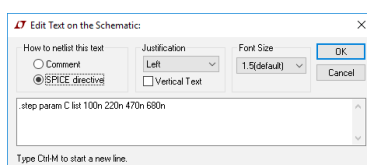
Rysunek 6. Schemat dla symulacji zasilacza LV POWER SUPPLY



Rysunek 7. Wynik symulacji zasilacza LV POWER SUPPLY



Rysunek 8. Użycie kursorów w symulacji zasilacza LV POWER SUPPLY



Rysunek 9. Parametryzowanie kondensatora C4 w symulacji zasilacza LV POWER SUPPLY

wyniki pokazane na **rysunku 10**. Widać, że kolejne pojemności kondensatora powodują coraz większe opóźnienie wyłączenia napięcia 9 V. Dla największej wartości kondensatora (680 nF) opóźnienie wzrasta do 330 ms. Z symulacji wynika, jak manipulując wartością kondensatora C4 można uzyskać różne czasy opóźnień wyłączenia napięcia 9 V. Przyjęto, że pojemność 100 nF jest wystarczające dla poprawnej pracy urządzenia. Niemniej jednak, gdyby w praktyce okazała się ona zbyt mała, to już wiemy, jakie inne pojemności możemy wziąć pod uwagę.

Sprawdźmy teraz kolejny element urządzenia: wejście AUX, a dokładniej jego charakterystykę częstotliwościową. Układ pokazano na **rysunku 11**. Rezystor R100 odpowiada impedancji wejściowej następnego stopnia. Przyjęto dwie możliwe wartości, aby przekonać się jak zmiana tej wartości wpływa na charakterystykę częstotliwości. Typ symulacji ustawiamy na „AC Analysis” w zakresie od 10 Hz do 1 MHz. Rezultat zaprezentowano na **rysunku 12**. Widać, że zmiana impedancji następnego stopnia wpływa minimalnie na wzmocnienie obwodu. Niemniej jednak, nawet tak nieznaczne zmiany mogą być łatwo skompensowane w układzie TDA7418. Częstotliwość graniczną filtra można znowu łatwo sprawdzić przy użyciu kursorów dostępnych w symulatorze. Wybierzmy jeden z wykresów (poprzez dopisanie ciągu znaków „@1” do wyświetlanego napięcia)

i włączmy obydwa kursory. Wynik pokazano na **rysunku 13**, a rezultat włączenia kursorów na **rysunku 14**. Częstotliwość graniczna filtra wynosi około 178 kHz, więc całkiem sporo, jak na układ tego typu. Pamiętajmy jedynie, że rzeczywiste pasmo może się jeszcze zmienić w zależności od wejściowej impedancji kolejnego stopnia.

Sprawdźmy teraz trochę bardziej skomplikowany układ: wyciszanie i włączanie końcówek mocy w momencie włączenia wzmacniacza i w przypadku użycia słuchawek (wtedy końcówki mocy mają zostać wyciszone). Układ testowy pokazano na **rysunku 15**. Włączanie słuchawek jest zasymulowane jako opornik R100 przyjmujący dwie wartości: 10 Ω i 10 M Ω , co odpowiada zwartym i rozwartym stykom gniazda słuchawkowego S i SS. Typ symulacji ustawiamy na „Transient” i czas trwania 2 sekundy. Chcemy wyświetlić prąd pobierany z układu LM3886 (w symulacji wyświetlamy prąd płynący przez R101). Wyniki przedstawiono na **rysunku 16**. Wykres zielony odpowiada pierwszej wartości opornika R100 (gniazdo PHONES zwarte), a wykres niebieski odpowiada drugiej wartości opornika R100 (wtyk słuchawek włożony w gniazdo). Napięcie zasilania układu jest wyłączane po upływie 1 sekundy, aby zbadać wyciszanie układów przy wyłączaniu urządzenia. W pierwszym wypadku prąd pobierany z wyprowadzenia MUTE układu LM3886 wynosi ponad 0,5 mA, a taka jest wartość oczekiwana i czas opóźnienia pojawienia się tego prądu wynosi 100 ms, co można odczytać z osi czasu. Po zaniku napięcia, prąd w ciągu kilkunastu milisekund maleje do wartości, która powinna wyciszyć końcówki mocy. Natomiast w drugim przypadku, prąd ten w ogóle się nie pojawia, co oznacza, że układy końcówek mocy są wyciszone (działa jedynie wzmacniacz słuchawkowy).

Sprawdźmy teraz stopień wyjściowy końcówek mocy. Schemat testowanego obwodu pokazano na **rysunku 17**. Sprawdzimy charakterystykę częstotliwościową, wzmocnienie i dobierzemy kondensator znajdujący się pomiędzy odwracającym i nieodwracającym wejściami LM3886. Typ symulacji ustawiamy na „AC Analysis” z zakresem częstotliwości 10 Hz...1 MHz. Przyjmijmy, że kondensator C50 ma 47 pF (wartość ustawiona jako parametr). Później zmienimy

DODATKOWE MATERIAŁY DO POBRANIA ZE STRONY:

www.media.avt.pl

W ofercie AVT*

AVT-5617 SAR

Podstawowe informacje:

- Wysokiej jakości tuner FM z obsługą komunikatów systemu RDS.
- Moduł Bluetooth obsługujący profil A2DP i kodek aptX zapewniający dźwięk klasy Hi-Fi.
- Obsługa dodatkowych wejść CD i AUX.
- 3-pasmowa regulacja barwy dźwięku.
- Wbudowany wzmacniacz słuchawkowy z niezależną regulacją głośności.
- Wzmacniacz mocy Hi-Fi, ok. 20 W/kanal.
- Obsługa pilota zdalnego sterowania.
- Obsługa trybu czuwania urządzenia (standby).
- Atrakcyjny, graficzny interfejs użytkownika.
- Podświetlana klawiatura na panelu sterującym.

Projekty pokrewne na www.media.avt.pl:

AVT-1973	Miniaturowy, stereofoniczny wzmacniacz mocy (EP 10/2017)
AVT-5602	Stereofoniczny wzmacniacz klasy D o mocy 2×50 W (EP 8/2017)
AVT-5547	4-kanalowy wzmacniacz mocy audio (EP 10/2016)
AVT-1934	Miniaturowy wzmacniacz mocy 2×1 W/8 Ω (EP 9/2016)
AVT-1923	Końcówka o mocy 2×60...100 W (EP 8/2016)
AVT-5540	Radioodbiornik dla każdego (EP 5/2016)
AVT-5528	Wzmacniacz audio klasy D o mocy do 2×50 W (EP 2/2016)
AVT-1843	Pamp_TDA7388 Wzmacniacz mocy audio 4×20 W/4 Ω (EP 2/2015)
AVT-1833	Pamp_LM4766 – wzmacniacz mocy audio 2×20 W/8 Ω (EP 12/2014)
AVT-5416	DAMP – wzmacniacz klasy D o mocy 10 W (EP 9/2013)
AVT-1758	Wzmacniacz z układem TPA3110 (EP 8/2013)
AVT-1746	Wzmacniacz o mocy 20 W z układem LM1875 (EP 7/2013)

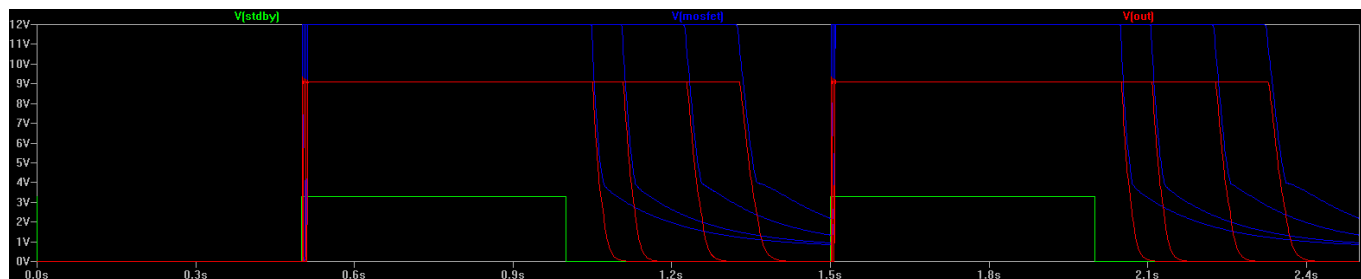
* Uwaga! Elektroniczne zestawy do samodzielnego montażu.

Wymagana umiejętność lutownia!

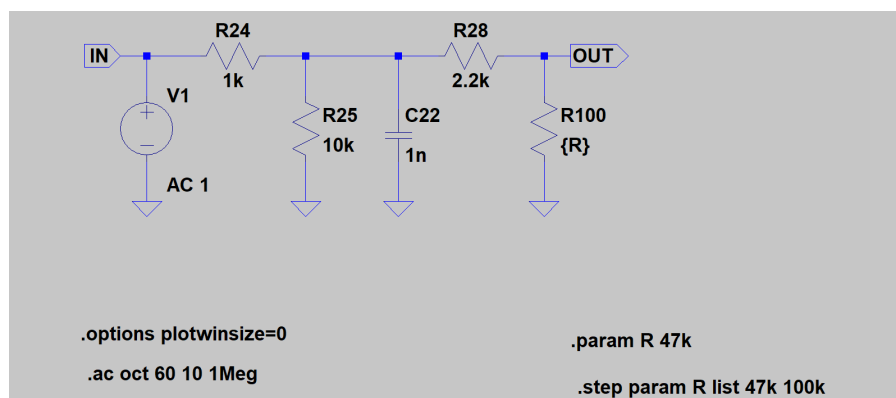
Podstawową wersją zestawu jest wersja [B] nazywana potocznie KiTem (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wylutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wylutowane w płytkę PCB)
- wersja [A] płytką drukowaną bez elementów i dokumentacją Kity w których występuje układ scalony wymagający zaprogramowania, posiadają następujące dodatkowe wersje:
- wersja [A+] płytką drukowaną [A] + zaprogramowany układ [UK] i dokumentacją
- 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://shop.avt.pl>



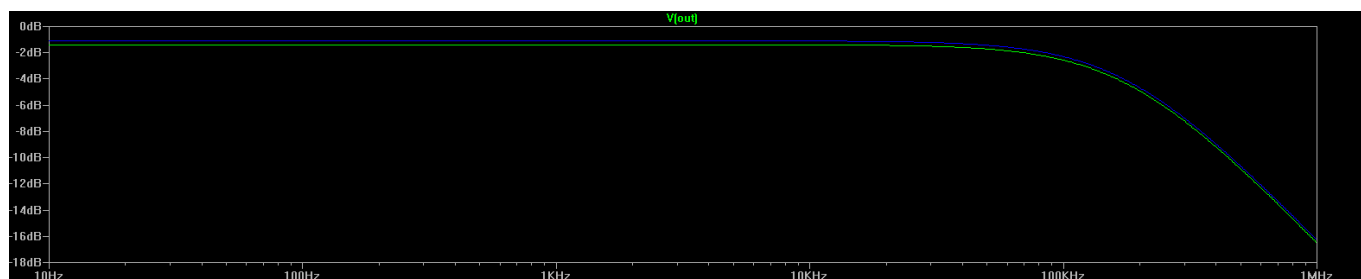
Rysunek 10. Wynik symulacji zasilacza LV POWER SUPPLY z parametryzowaniem wartości kondensatora C4



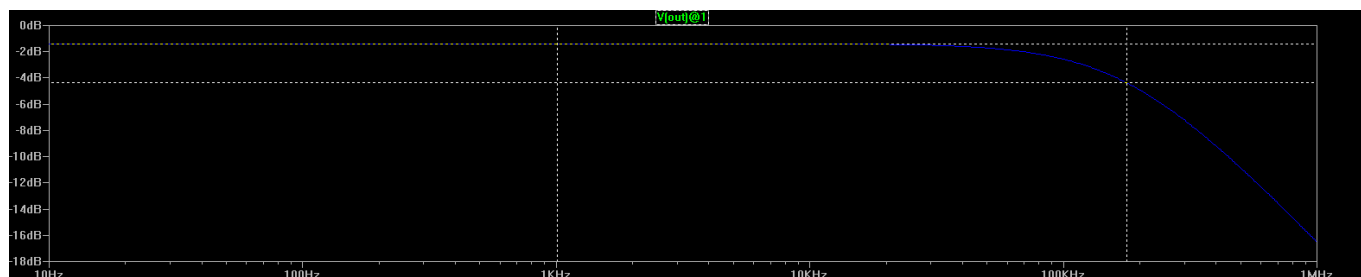
Rysunek 11. Schemat dla symulacji wejścia AUX

to na listę wartości. Wynik symulacji pokazano na **rysunku 18**. Wzmocnienie wynosi 26,5 dB, a przy częstotliwości 10 Hz występuje minimalny spadek wzmocnienia. Pasmo przenoszenia jest dość szerokie. Aby je sprawdzić włączamy kursory. Wynik jest dość zaskakujący – pasmo jest bardzo szerokie (**rysunek 19**). Prawdopodobnie nie jest to prawdziwa wartość i być może wynika z niedokładności modelu LM3886. Pozostaje jedynie sprawdzić, jak to wygląda po zmontowaniu rzeczywistego układu.

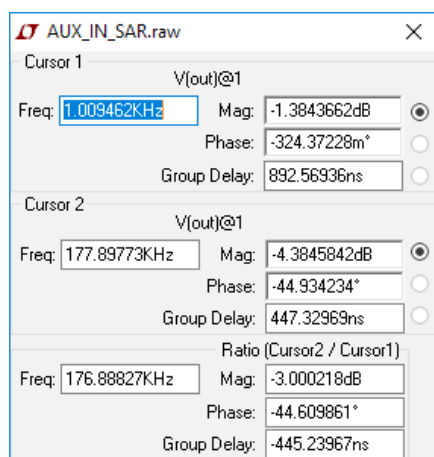
Sprawdźmy teraz, jaki jest efekt użycia różnych wartości kondensatora na wejściach



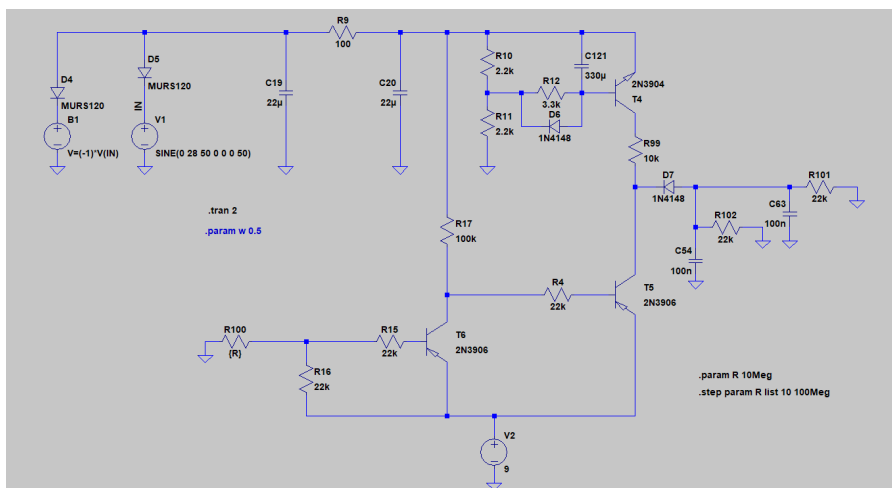
Rysunek 12. Wynik symulacji wejścia AUX

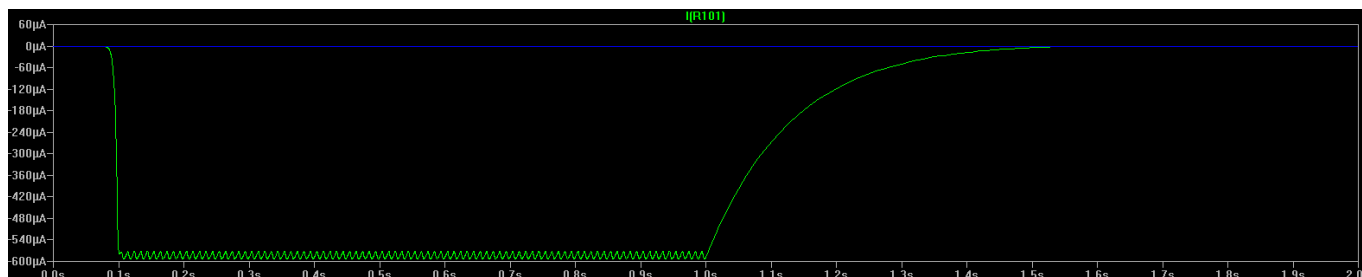


Rysunek 13. Ustawienia kursorów dla symulacji wejścia AUX

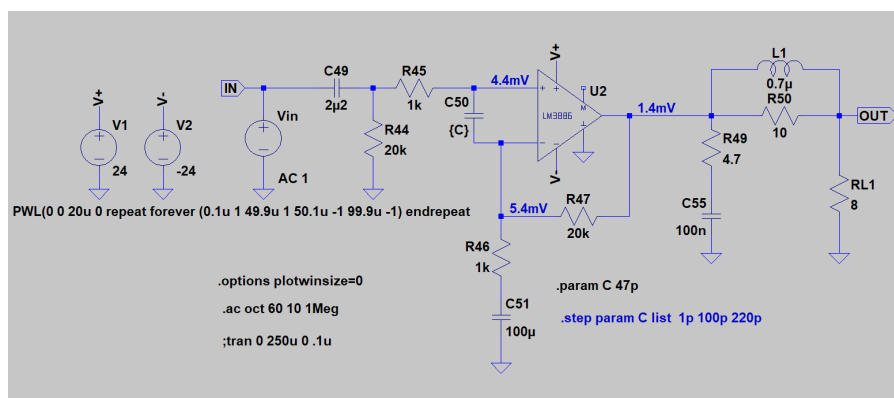


Rysunek 14. Rezultat włączenia kursorów dla symulacji wejścia AUX





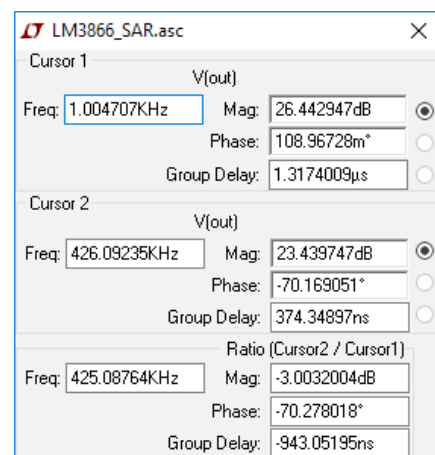
Rysunek 16. Wynik symulacji układu wyciszania i włączania końcówek mocy



Rysunek 17. Schemat dla symulacji stopnia wyjściowego końcówek mocy

LM3886. Producent układu zaleca wartość 220 pF. Potrzeba użycia tego kondensatora wynika z konieczności zmniejszenia wzmocnienia układów przy dużych częstotliwościach, ale jednocześnie karta katalogowa wspomina, że większa wartość kondensatora może mieć negatywne skutki. Zobaczmy, jak to wygląda w symulacji. Uaktywnijmy opatrzoną komentarzem dyrektywę z listą wartości tego kondensatora i uruchommy symulację. Wyniki pokazano na **rysunku 20**. Największa pojemność kondensatora (220 pF) powoduje podwyższenie wzmocnienia układu przy częstotliwości 277 kHz, co może prowadzić do wzbudzenia się układu. Aby tego uniknąć, przyjęto dość konserwatywną pojemność wynoszącą 47 pF. Jako wniosek

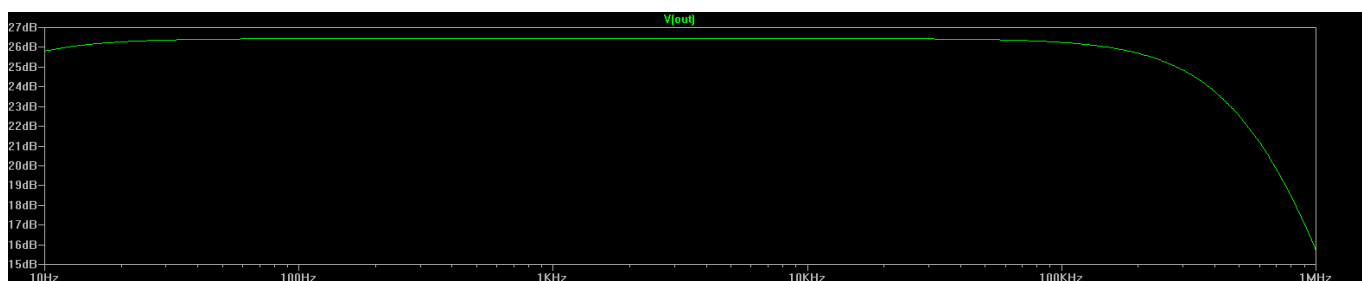
z przeprowadzonych symulacji można przyjąć, że pomagają one w projektowaniu układów i pozwalają na podejmowanie decyzji nawet bez fizycznej budowy układów. Niemniej jednak, dopiero rzeczywiste wykonanie układu (i sprawdzenie jego działania) jest ostatecznym dowodem na poprawność jego implementacji. Symulacje w pewnych przypadkach mogą dawać niezbyt dokładne wyniki. W większości wypadków przyczyną takiej sytuacji są niezbyt dokładne modele symulowanych układów scalonych i tranzystorów. Tym niemniej, użycie symulatora znacznie przyspiesza i ułatwia projektowanie. Zresztą obecnie tak właśnie projektuje się układy analogowe. Jest to również bardzo użyteczna metoda przy naprawie sprzętów audio.



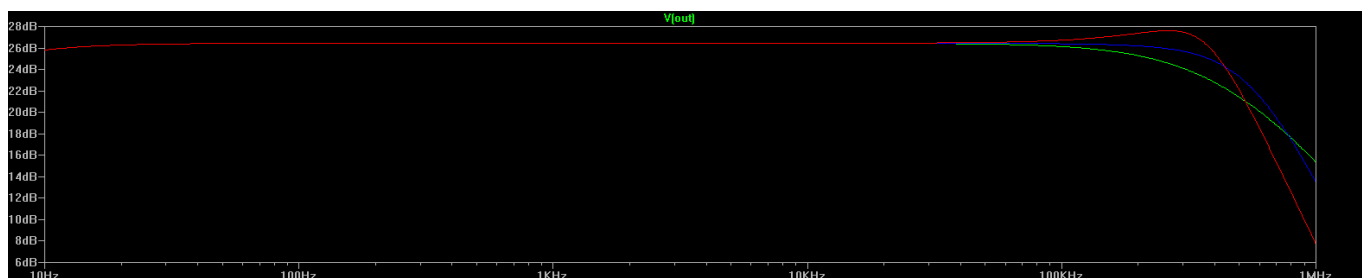
Rysunek 19. Ustawienia kursorów w symulacji stopnia wyjściowego końcówek mocy

Moduł Bluetooth Flaircomm BTM501VQ1C

Kończąc opis płyty głównej amplitunera SAR pozostało mi przybliżyć nieco kwestię zastosowanego modułu Bluetooth, jako że jest to bardzo ciekawy podzespół. Wspomniany moduł jest urządzeniem drugiej klasy mocy produkowanym przez firmę Flaircomm, model BTM501VQ1C, zgodnym ze standardem BT 2.1+EDR i wspierającym większość profili dostępnych w ramach tego standardu, a mianowicie HS/HF, A2DP, AVRCP, OPP, DUN i SPP. Moduł obsługuje zestaw komend AT zgodny ze standardami GSM 07.05



Rysunek 18. Wynik symulacji stopnia wyjściowego końcówek mocy



Rysunek 20. Wynik symulacji stopnia wyjściowego końcówek mocy z parametryzowaniem wartości kondensatora na wejściu układu LM3886

i GSM07.07 oraz z uwagi na swoją rozbudowaną funkcjonalność, dodatkowe komendy wprowadzone przez producenta podzespołu. Komunikacja odbywa się za pomocą interfejsu UART z maksymalną prędkością transmisji wynoszącą aż 4 Mb/s. Ponadto, ma również interfejsy SPI oraz USB, które mogą znaleźć zastosowanie w wypadku zmiany oprogramowania sterującego.

Moduł BT wyposażono w monofoniczne wejście mikrofonowe oraz stereofoniczne, różnicowe wyjście głośnikowe. Oba przetworniki (A/D i D/A) zintegrowane w module obsługują następujące częstotliwości próbkowania: 8 kHz; 11,025 kHz; 16 kHz; 22,05 kHz; 24 kHz; 32 kHz; 44,1 kHz i 48 kHz. Opis modułu ograniczę do przedstawienia podstawowych informacji dotyczących uruchomienia, konfiguracji i nawiązania połączenia A2DP.

Komunikacja z modulem odbywa się poprzez interfejs USART z domyślną prędkością równą 115 kbps. Konfiguracja i obsługa modułu polegają na wysyłaniu do niego odpowiednich do wykonywanych czynności komend AT i odbieraniu odpowiedzi, które sygnalizują stan wykonania rozkazu lub stan pracy urządzenia. Komendy wysyłane do modemu mają postać:

AT#CMD<CR><LF>

AT#CMD sp <CR><LF>, gdzie:

AT# identyfikator rozszerzonej komendy firmy Flaircomm,

CMD identyfikator komendy,

sp opcjonalny parametr wysyłanej komendy,

<CR><LF> znaki końca linii i powrotu karetki.

W odpowiedzi na wysłaną komendę modemu Bluetooth wysła potwierdzenie jej wykonania w postaci ciągu znaków „OK<CR><LF>” w razie powodzenia lub „ERROR<CR><LF>” w razie wystąpienia błędu. Ponadto, może wysyłać różnego rodzaju odpowiedź („Indication”), które sygnalizują stan pracy urządzenia, a których to składnia przedstawia się następująco:

IND<CR><LF>

IND sp <CR><LF>, gdzie:

IND identyfikator zdarzenia,

sp opcjonalny parametr związany ze zdarzeniem.

To tyle, w ekspresowym skrócie, jeśli chodzi o podstawowe informacje dotyczące sposobu komunikacji z modulem. Przejdę do zagadnień programowych zaczynając, jak zwykle, od przedstawienia pliku nagłówkowego do obsługi modułu w zakresie profilu A2DP, który to pokazano na **listingu 6**. Zdefiniowano w nim zmienną globalną przeznaczoną do przechowywania statusu modułu Bluetooth. Ponadto, w pliku obsługi zdefiniowano szereg stałych znakowych reprezentujących rozkazy sterujące oraz odpowiedzi modułu. Stałe te zamieszczono na **listingu 7**.

Listing 1. Zawartość pliku nagłówkowego do obsługi układu TDA7418

```
typedef struct
{
    volatile uint8_t Input; //0...3 -> PD/SE4, SE1...SE3
    volatile uint8_t frontVolume; //0...79 -> -79dB...0dB
    volatile uint8_t rearVolume; //0...79 -> -79dB...0dB
    volatile uint8_t Bass; //0...30 -> -15dB...+15dB
    volatile uint8_t Middle; //0...30 -> -15dB...+15dB
    volatile uint8_t Treble; //0...30 -> -15dB...+15dB
} TDA7418config;

extern TDA7418config TDA7418; //Zmienna przechowująca interesujące nas parametry TDA7418

#define TDA7418_WR_ADDR 0x88
#define TDA7418_RD_ADDR 0x89
//Option
#define TEST_MODE_OFF (0<<7)
#define TEST_MODE_ON (1<<7)
#define AUTO_ZERO_REMAIN_OFF (0<<6)
#define AUTO_ZERO_REMAIN_ON (1<<6)
#define AUTO_INCR_MODE_OFF (0<<5)
#define AUTO_INCR_MODE_ON (1<<5)
//Registers and options
#define REG_SOURCE_SELECTOR 0x00
#define SOURCE_SE4 0x00
#define SOURCE_SE1 0x01
#define SOURCE_SE2 0x02
#define SOURCE_SE3 0x03
#define SINGLE_ENDED (0<<7)
#define DIFFERENTIAL_STEREO (1<<7)
/**/
#define REG_LOUDNESS 0x01
#define CENTER_FREQ_FLAT (0<<4)
#define CENTER_FREQ_400HZ (1<<4)
#define CENTER_FREQ_800HZ (2<<4)
#define CENTER_FREQ_2400HZ (3<<4)
#define LOW_BOST (0<<6)
#define LOW_HIGH_BOST (1<<6)
#define LOUDNESS_SOFT_STEP_ON (0<<7)
#define LOUDNESS_SOFT_STEP_OFF (1<<7)
/**/
#define REG_VOLUME 0x02
#define VOLUME_SOFT_STEP_ON (0<<7)
#define VOLUME_SOFT_STEP_OFF (1<<7)
/**/
#define REG_TREBLE 0x03
#define TREBLE_CENTER_FREQ_10KHZ (4<<5)
#define TREBLE_CENTER_FREQ_12_5KHZ (5<<5)
#define TREBLE_CENTER_FREQ_15KHZ (6<<5)
#define TREBLE_CENTER_FREQ_17_5KHZ (7<<5)
/**/
#define REG_MIDDLE 0x04
#define MID_Q_FACT_0_5 (0<<5)
#define MID_Q_FACT_0_75 (1<<5)
#define MID_Q_FACT_1 (2<<5)
#define MID_Q_FACT_1_25 (3<<5)
#define MID_SOFT_STEP_ON (0<<7)
#define MID_SOFT_STEP_OFF (1<<7)
/**/
#define REG_BASS 0x05
#define BASS_Q_FACT_1 (0<<5)
#define BASS_Q_FACT_1_25 (1<<5)
#define BASS_Q_FACT_1_5 (2<<5)
#define BASS_Q_FACT_2 (3<<5)
#define BASS_SOFT_STEP_ON (0<<7)
#define BASS_SOFT_STEP_OFF (1<<7)
/**/
#define REG_MID_BAS_CF 0x06
#define MID_CENTER_FREQ_500HZ (0<<0)
#define MID_CENTER_FREQ_1KHZ (1<<0)
#define MID_CENTER_FREQ_1_5KHZ (2<<0)
#define MID_CENTER_FREQ_2_5KHZ (3<<0)
#define BASS_CENTER_FREQ_60HZ (0<<2)
#define BASS_CENTER_FREQ_80HZ (1<<2)
#define BASS_CENTER_FREQ_100HZ (2<<2)
#define BASS_CENTER_FREQ_200HZ (3<<2)
#define BASS_DC_MODE_OFF (0<<4)
#define BASS_DC_MODE_ON (1<<4)
#define SMOOTHING_FILTER_OFF (0<<5)
#define SMOOTHING_FILTER_ON (1<<5)
/**/
#define REG_SPK_ATT_FL 0x07
#define REG_SPK_ATT_RL 0x08
#define REG_SPK_ATT_RR 0x09
#define REG_SPK_ATT_FR 0x0A
#define REG_SUBB_ATT 0x0B
#define SPEAKER_MUTED 0x60
/**/
#define REG_SOFT_MUTE 0x0C
#define SOFT_MUTE_ON (0<<0)
#define SOFT_MUTE_OFF (1<<0)
#define SOFT_MUTE_TIME_48MS (0<<1)
#define SOFT_MUTE_TIME_96MS (1<<1)
#define SOFT_MUTE_TIME_123MS (2<<1)
#define SOFT_STEP_TIME_0_16MS (0<<3)
#define SOFT_STEP_TIME_0_32MS (1<<3)
#define SOFT_STEP_TIME_0_64MS (2<<3)
#define SOFT_STEP_TIME_1_28MS (3<<3)
#define SOFT_STEP_TIME_2_56MS (4<<3)
#define SOFT_STEP_TIME_5_12MS (5<<3)
#define SOFT_STEP_TIME_10_24MS (6<<3)
#define SOFT_STEP_TIME_20_48MS (7<<3)
#define AUTO_ZERO_OFF (0<<6)
#define AUTO_ZERO_ON (1<<6)
/**/
#define REG_TEST 0x0D
```

Listing 2. Funkcja inicjalizacji układu TDA7418

```
void TDA7418init(void)
{
    //Konfigurujemy układ TDA7418 wpisując wartości do kolejnych rejestrów w trybie autoinkrementacji
    TWIstart();
    TWIwriteByte(TDA7418.WR_ADDR);
    TWIwriteByte(REG_SOURCE_SELECTOR|AUTO_INCR_MODE_ON);
    //Input Gain zależny od wejścia, Single Ended Stereo
    switch(TDA7418.Input)
    {
        case INPUT_FM: TWIwriteByte(TDA7418.Input|6<<3); break; //6 dB
        case INPUT_BT: TWIwriteByte(TDA7418.Input|8<<3); break; //8 dB
        default: TWIwriteByte(TDA7418.Input); //0 dB
    }
    TWIwriteByte(CENTER_FREQ_FLAT|LOW_BOST|LOUDNESS_SOFT_STEP_ON); //Loudness: Attenuation = 0dB, Flat, Low Boost, Soft Step=On
    TWIwriteByte(0x00); //Master Volume: Attenuation = 0dB, Soft Step=On
    TWIwriteByte((TDA7418.Treble>15? 46-TDA7418.Treble:TDA7418.Treble)|TREBLE_CENTER_FREQ_12_5KHZ); //Treble
    TWIwriteByte((TDA7418.Middle>15? 46-TDA7418.Middle:TDA7418.Middle)|MID_Q_FACT_1); //Middle
    TWIwriteByte((TDA7418.Bass>15? 46-TDA7418.Bass:TDA7418.Bass)|BASS_Q_FACT_1); //Bass
    TWIwriteByte(BASS_CENTER_FREQ_80HZ|MID_CENTER_FREQ_1_5KHZ|SMOOTHING_FILTER_ON);
    TWIwriteByte(95-TDA7418.FrontVolume); //Front Left Attenuation, Soft Step=On
    TWIwriteByte(95-TDA7418.RearVolume); //Rear Left Attenuation, Soft Step=On
    TWIwriteByte(95-TDA7418.RearVolume); //Rear Right Attenuation, Soft Step=On
    TWIwriteByte(95-TDA7418.FrontVolume); //Front Right Attenuation, Soft Step=On
    TWIwriteByte(SPEAKER_MUTED); //Subwoofer Attenuation = Mute, Soft Step=On
    TWIwriteByte(SOFT_MUTE_OFF|SOFT_MUTE_TIME_48MS|SOFT_STEP_1_28MS|AUTO_ZERO_ON);
    TWIwriteByte(0x00); //Audio Processor Testing Mode=Off
    TWIstop();
}
```

Na początek potrzebne będą funkcje, dzięki którym stanie się możliwa komunikacja przez interfejs USART – pokazano je na **listingu 8**. Myślę, że z uwagi na nieskomplikowaną składnię i nazewnictwo sugerujące pełnioną funkcję nie wymagają dodatkowego komentarza. Kolejna funkcja

to funkcja narzędziowa, dzięki której możemy wysłać rozkaz do modułu Bluetooth wraz z towarzyszącym mu parametrem. Tę funkcję pokazano na **listingu 9**. Dalej, funkcja przeznaczona do uruchomienia i przeprowadzenia podstawowej konfiguracji modułu – pokazano ją na **listingu 10**. I na koniec,

na **listingu 11** funkcja obsługi przerwania od odebrania znaku interfejsu USART (USART0_RX_vect) odpowiedzialna za obsługę odpowiedzi wysyłanych przez moduł Bluetooth.

Dobór radiatora

Pozostaje dość ważna kwestia związana z opisywanym modulem płyty głównej, a mianowicie kwestia doboru parametrów transformatora mocy oraz chłodzenia układów LM3886. Jak z pewnością wicie, kluczową sprawą przy konstruowaniu jakiegokolwiek wzmacniacza mocy są wymogi dotyczące źródła napięcia zasilania końcówki mocy. Z pozoru wydawałoby się, iż jest to zadanie dość proste,

Listing 3. Funkcja odpowiedzialna za wybór aktywnego wejścia selektora wejściowego układu TDA7418

```
void TDA7418setInput(void)
{
    TWIstart();
    TWIwriteByte(TDA7418.WR_ADDR);
    TWIwriteByte(REG_SOURCE_SELECTOR|AUTO_ZERO_REMAIN_OFF);
    //Input Gain zależny od wejścia, Single Ended Stereo
    switch(TDA7418.Input)
    {
        case INPUT_FM: TWIwriteByte(TDA7418.Input|6<<3); break; //6 dB
        case INPUT_BT: TWIwriteByte(TDA7418.Input|8<<3); break; //8 dB
        default: TWIwriteByte(TDA7418.Input); //0 dB
    }
    TWIstop();
}
```

Listing 4. Funkcja odpowiedzialna za regulację tonów układu TDA7418

```
void TDA7418setTone(uint8_t toneReg)
{
    TWIstart();
    TWIwriteByte(TDA7418.WR_ADDR);
    TWIwriteByte(toneReg|AUTO_ZERO_REMAIN_ON);
    switch(toneReg)
    {
        case REG_TREBLE: TWIwriteByte((TDA7418.Treble>15? 46-TDA7418.Treble:TDA7418.Treble)|TREBLE_CENTER_FREQ_12_5KHZ); break;
        case REG_MIDDLE: TWIwriteByte((TDA7418.Middle>15? 46-TDA7418.Middle:TDA7418.Middle)|MID_Q_FACT_1); break;
        case REG_BASS: TWIwriteByte((TDA7418.Bass>15? 46-TDA7418.Bass:TDA7418.Bass)|BASS_Q_FACT_1); break;
    }
    TWIstop();
}
```

Listing 5. Funkcja odpowiedzialna za regulację głośności układu TDA7418

```
void TDA7418setVolume(void)
{
    TWIstart();
    TWIwriteByte(TDA7418.WR_ADDR);
    TWIwriteByte(REG_SPK_ATT_FL|AUTO_INCR_MODE_ON|AUTO_ZERO_REMAIN_ON);
    TWIwriteByte(95-TDA7418.FrontVolume); //Front Left Attenuation, Soft Step=On
    TWIwriteByte(95-TDA7418.RearVolume); //Rear Left Attenuation, Soft Step=On
    TWIwriteByte(95-TDA7418.RearVolume); //Rear Right Attenuation, Soft Step=On
    TWIwriteByte(95-TDA7418.FrontVolume); //Front Right Attenuation, Soft Step=On
    TWIstop();
}
```

Listing 6. Plik nagłówkowy do obsługi modułu Bluetooth w zakresie profilu A2DP

```
//Prędkość interfejsu
#define USART_BAUD 115200
#define __UBRR (F_CPU/16/USART_BAUD-1)
//Maksymalna liczba bajtów w odpowiedzi
#define MAX_BYTES 32
//Definicje statusów interfejsu BT
#define BT_DISCONNECTED 0x00
#define BT_CONNECTED 0x01
#define BT_MUSIC_PAUSE 0x02
#define BT_MUSIC_PLAY 0x03
//Znaczniki ASCII
#define ASCII_LF 10
#define ASCII_CR 13
//Zmienne modułu
extern volatile uint8_t BTstatus; //Flaga statusu interfejsu BT
```

w rzeczywistości okazuje się, że podczas procesu projektowania należy wziąć pod uwagę wiele kluczowych kwestii by projektowany przez nas wzmacniacz pracował według wstępnych założeń. Jak łatwo się domyślić, kluczową sprawą jest tutaj moc i napięcie wyjściowe zastosowanego transformatora mocy stanowiącego źródło napięcia zasilania końcówkę mocy. Zbyt niskie napięcie wyjściowe transformatora ograniczy nam maksymalną moc konstruowanej końcówki mocy, zaś zbyt niska jego moc powodować będzie zniekształcenia przy wyższych poziomach głośności. Przyznam szczerze, iż poszukując fachowej literatury w tym zakresie natknąłem się na doskonałe kompendium wiedzy pod postacią strony internetowej znajdującej się pod adresem <https://goo.gl/91fLh4>, której lekturę szczególnie polecam!

Proces wyznaczania parametrów transformatora mocy rozpoczynamy od obliczenia szczytowego napięcia na wyjściu końcówki mocy V_{opeak} , przy założonej impedancji obciążenia i maksymalnej, projektowanej mocy szczytowej. Wartość tę wyznaczamy według następującego wzoru:

$$V_{opeak} = \sqrt{2 \cdot R_L \cdot P_O}, \text{ gdzie:}$$

- R_L – impedancja obciążenia (głośników),
- P_O – projektowana moc szczytowa wzmacniacza.

Dla naszych, założonych danych projektowych otrzymujemy:

$$V_{opeak} = \sqrt{2 \cdot 8\Omega \cdot 20W} = 17,9V$$

Wiedząc już, jakie napięcie szczytowe wystąpi na zaciskach końcówki mocy przy maksymalnej mocy wyjściowej możemy wyznaczyć niezbędną wartość napięcia zasilającego końcówkę mocy V_{supply} dla uzyskania wspomnianych parametrów wyjściowych. Napięcie to wyznaczamy według wzoru:

$$V_{supply} = \pm(V_{opeak} + V_{drop}) \cdot (1 + C_{trafo}) \cdot C_{reg},$$

gdzie:

- V_{drop} – spadek napięcia na końcówce mocy LM3886 (4V według dokumentacji producenta),
- C_{trafo} – współczynnik stanu jałowego transformatora, określający o ile wzrośnie napięcie uzwojenia wtórnego w stanie jałowym (jeśli nie dysponujemy tym parametrem, dla transformatorów toroidalnych możemy przyjąć wartość 0,1 odpowiadającą wzrostowi napięcia o 10%)
- C_{reg} – współczynnik określający zakres zmian napięcia sieci zasilającej 230 V, który wynosi 1,1 (inaczej 10%).

Mając wszystkie, niezbędne wartości obliczamy:

$$V_{supply} = \pm(17,9V + 4V) \cdot (1 + 0,1) \cdot 1,1 = \pm 26,5V$$

Wynika z tego, że aby na wyjściu wzmacniacza mocy dostarczyć moc równą

Listing 7. Lista stałych związanych z obsługą modułu Bluetooth

```
//Definicje obsługiwanych rozkazów sterujących i odpowiedzi modułu Flaircomm BTM-501
const char cmd_ENTER_PAIRING_MODE[] PROGMEM = "AT#CA";
const char cmd_CANCEL_PAIRING_MODE[] PROGMEM = "AT#CB";
const char cmd_PLAY_PAUSE_MUSIC[] PROGMEM = "AT#MA";
const char cmd_STOP_MUSIC[] PROGMEM = "AT#MC";
const char cmd_FORWARD_MUSIC[] PROGMEM = "AT#MD";
const char cmd_BACKWARD_MUSIC[] PROGMEM = "AT#ME";
const char cmd_CHANGE_DEVICE_NAME[] PROGMEM = "AT#MM";
const char cmd_POWER_ON_MODULE[] PROGMEM = "AT#MW";
const char cmd_GET_PLAY_STATUS[] PROGMEM = "AT#QD";
const char cmd_GET_ELEMENT_ATTRIBUTES[] PROGMEM = "AT#QE";
const char cmd_ACCEPT_PAIRING_REQUEST[] PROGMEM = "AT#QJ";
const char cmd_REJECT_PAIRING_REQUEST[] PROGMEM = "AT#QK";
const char cmd_ALWAYS_DISCOVERABLE[] PROGMEM = "AT#QP";

const char ind_ENTER_PAIRING_MODE[] PROGMEM = "II";
const char ind_EXIT_PAIRING_MODE[] PROGMEM = "IJ";
const char ind_INIT_COMPLETE[] PROGMEM = "IS";
const char ind_AV_STOP[] PROGMEM = "MA";
const char ind_AV_PLAY[] PROGMEM = "MB";
const char ind_ELEMENT_ATTRIBUTES[] PROGMEM = "MH";
const char ind_PLAY_STATUS[] PROGMEM = "MJ";
const char ind_REPORT_DEVICE_NAME[] PROGMEM = "MM";
const char ind_A2DP_CONNECTED[] PROGMEM = "MK";
const char ind_A2DP_DISCONNECTED[] PROGMEM = "MY";
const char ind_A2DP_LINK_LOSS[] PROGMEM = "PP";
const char ind_PAIRING_REQUEST[] PROGMEM = "QA";
const char ind_PAIRING_SUCCESSFUL[] PROGMEM = "QB";
const char ind_PAIRING_FAILURE[] PROGMEM = "QC";
```

Listing 8. Funkcje obsługi USART mikrokontrolera

```
void USARTinit(void)
{
    //Ustawienie prędkości 115200 bps
    UBRR0H = (uint8_t)UBRR>>8; UBRR0L = (uint8_t)UBRR;
    //Załączenie nadajnika i odbiornika oraz uruchomienie przerwania od RX
    UCSRB |= (1<<RXEN0)|(1<<TXEN0)|(1<<RXCIE0);
    //Ustawienie formatu ramki: 8bitów danych, 1 bit stopu
    UCSRC = (3<<UCSZ00);
}

inline void USARTsendByte(uint8_t Byte)
{
    //Czekamy, aż bufor nadawczy będzie pusty
    while (!(UCSR0A & (1<<UDRE0)));
    UDR0 = Byte;
}

void USARTsendString_P(const char *String)
{
    register char Byte;
    //Wysyłamy łańcuch z pamięci FLASH dopóki nie napotkamy 0
    while ((Byte = pgm_read_byte(String++)) != 0) USARTsendByte(Byte);
}
```

Listing 9. Funkcja przeznaczona do wysłania rozkazu z parametrem do modułu Bluetooth

```
void BTsendCommand(const char *Command, const char *Parameters)
{
    USARTsendString_P(Command);
    USARTsendString_P(Parameters);
    USARTsendString_P(PSTR("\r\n"));
}
```

Listing 10. Funkcja do konfiguracji modułu Bluetooth

```
void BTconfig(void)
{
    BTsendCommand(cmd_POWER_ON_MODULE, NULL);
    BTsendCommand(cmd_ENTER_PAIRING_MODE, NULL);
    //Zmieniamy domyślną nazwę BT
    BTsendCommand(cmd_CHANGE_DEVICE_NAME, PSTR("eagleAudio SAR"));
}
```

Listing 11. Funkcja obsługi przerwania od odebrania znaku USART

```
/* Funkcja obsługuje interfejs USART0, czyli odbiór danych wysyłanych przez modem. Ramka: Odpowiedź<CR><LF> */
ISR(USART0_RX_vect)
{
    static char String[MAX_BYTES]; //Tablica przechowująca odpowiedź modemu
    static uint8_t Idx; //Indeks bieżącego elementu
    char recByte = UDR0; //Odczytanie odebranego bajta danych
    switch(recByte)
    {
        case ASCII_CR:
            break;
        case ASCII_LF:
            //Odebraliśmy kompletną odpowiedź modemu, więc ją analizujemy
            String[Idx] = '\0'; //Na końcu C-stringa umieszczamy NULL
            if((strcmp_P(String, ind_HFP_DISCONNECTED)==0) || (strcmp_P(String, ind_A2DP_DISCONNECTED)==0)) BTstatus = BT_DISCONNECTED;
            else if((strcmp_P(String, ind_HFP_CONNECTED)==0) || (strcmp_P(String, ind_A2DP_CONNECTED)==0)) BTstatus = BT_CONNECTED;
            else if(strcmp_P(String, ind_AV_STOP) == 0) BTstatus = BT_MUSIC_PAUSE;
            else if(strcmp_P(String, ind_AV_PLAY) == 0) BTstatus = BT_MUSIC_PLAY;

            Idx = 0;
            break;
        default:
            String[Idx++] = recByte;
    }
}
```

20 W przy impedancji obciążenia 8Ω należy dysponować źródłem napięcia zasilania o napięciu wyjściowym rzędu $\pm 26,5 \text{ V}$. Mając tę wartość przechodzimy do ostatniego kroku, czyli wyznaczenia napięcia wyjściowego uzwojenia wtórnego transformatora, którą to obliczamy według wzoru:

$$V_{\text{trafo}} = \frac{V_{\text{supply}}}{\sqrt{2} \cdot C_{\text{reg}} \cdot (1 + C_{\text{trafo}})} = \frac{26,5 \text{ V}}{\sqrt{2} \cdot 1,1 \cdot (1 + 0,1)} = 15,5 \text{ V}$$

Znamy już niezbędną wartość napięcia wtórnego transformatora mocy, w związku z czym możemy przejść do obliczeń związanych z mocą źródła napięcia zasilania. Na początek wyznaczmy niezbędną, szczytową moc źródła napięcia zasilania końcówki mocy, którą to wartość obliczamy według następującego wzoru:

$$P_{\text{supply}} = 2 \cdot V_{\text{supply}} \cdot \left(\frac{V_{\text{peak}}}{\pi \cdot R_L} + QPSC \right), \text{ gdzie}$$

- $QPSC$ – całkowity prąd spoczynkowy końcówki mocy LM3886 (85mA według dokumentacji producenta).

Mając wszystkie, niezbędne wartości obliczamy:

$$P_{\text{supply}} = 2 \cdot 26,5 \text{ V} \cdot \left(\frac{17,9 \text{ V}}{\pi \cdot 8 \Omega} + 0,085 \text{ A} \right) = 42 \text{ W}$$

Stąd już prosta droga do wyznaczenia znamionowej mocy transformatora zasilającego końcówkę mocy, którą to w przypadku wzmacniacza stereofonicznego obliczamy według wzoru:

$$P_{\text{trafo}} = 2 \cdot P_{\text{supply}} \cdot C_{\text{safe}}, \text{ gdzie}$$

- C_{safe} – współczynnik zapasu mocy równy 1.5 (zwyczajowo).

Moc naszego transformatora powinna w takim wypadku wynosić

$$P_{\text{trafo}} = 2 \cdot 42 \cdot 1,5 = 126 \text{ VA}$$

Interesujące, nieprawdaż? Wynika z tego, iż aby dostarczyć do głośników o impedancji 8Ω moc $2 \times 20 \text{ W}$ musimy dysponować transformatorem toroidalnym o napięciu wyjściowym $2 \times 15,5 \text{ V}$ i mocy aż 126 VA ! Warto o tym pamiętać przyglądając się niewielkim zestawom stereo, które opatrzone niejednokrotnie opisem $2 \times 100 \text{ W}$ lub nawet więcej. Wracając do naszego urządzenia, niezbędną moc transformatora zasilającego obliczono przy założeniu, iż na wejście wzmacniacza mocy dostarczamy sygnał sinusoidalny. Jako, że typowy sygnał dźwiękowy charakteryzuje się dużą zmiennością amplitudy i częstotliwości, a dodatkowo transformatory toroidalne mają sporą możliwość chwilowego przeciążania, zastosujemy transformator o mocy 100 VA , moim zdaniem w zupełności wystarczający.

Ostatnią czynnością przy konstruowaniu wzmacniacza mocy, jest dobór odpowiedniego elementu rozpraszającego straty mocy układu LM3886, a więc dobór odpowiedniego radiatora. Dobór ten rozpoczynamy od wyznaczenia maksymalnej mocy rozpraszanej P_{dmax} przez układy LM3886, którą to wartość obliczamy zgodnie ze wzorem:

$$P_{\text{dmax}} = \frac{(2 \cdot V_{\text{supply}})^2}{2 \cdot \pi^2 \cdot R_L} = \frac{(2 \cdot 26,5 \text{ V})^2}{2 \cdot \pi^2 \cdot 8 \Omega} = 17,8 \text{ W}$$

Wynika z tego, że dla zapewnienia stabilnej pracy końcówki mocy, przy maksymalnej mocy dostarczanej do głośników, nasz radiator musi być zdolny do rozproszenia mocy strat rzędu $17,8 \text{ W}$. Aby było to możliwe, element ten musi charakteryzować się odpowiednią rezystancją termiczną Θ_{sa} . Co więcej, na drodze pomiędzy krzemową strukturą końcówki mocy układu LM3886 a otoczeniem, do którego to zostanie rozproszona ta moc, znajdują się jeszcze inne elementy wpływające na zdolność rozpraszania

ciepła, w związku z czym wzór na maksymalną rezystancję termiczną zastosowanego radiatora uwzględniający je obliczamy zgodnie ze wzorem:

$$\Theta_{sa} = \frac{(T_{jmax} - T_{amb}) - P_{dmax} \cdot (\Theta_{jc} + \Theta_{cs})}{P_{dmax}}$$

gdzie

- T_{jmax} – maksymalna, dopuszczalna temperatura struktury krzemowej (150°C według dokumentacji producenta),
- T_{amb} – obliczeniowa temperatura otoczenia,
- Θ_{jc} – rezystancja termiczna pomiędzy strukturą krzemową a obudową układu LM3886, różna w zależności od wersji układu (z izolowaną: 2°C/W lub nieizolowaną obudową: 1°C/W),
- Θ_{cs} – rezystancja termiczna pomiędzy obudową układu LM3886 a radiator (rezystancja termiczna silikonowej podkładki izolacyjnej lub pasty termicznej).

Mając wszystkie, niezbędne wartości obliczamy maksymalną rezystancję termiczną, jaką musi charakteryzować się zastosowany radiator:

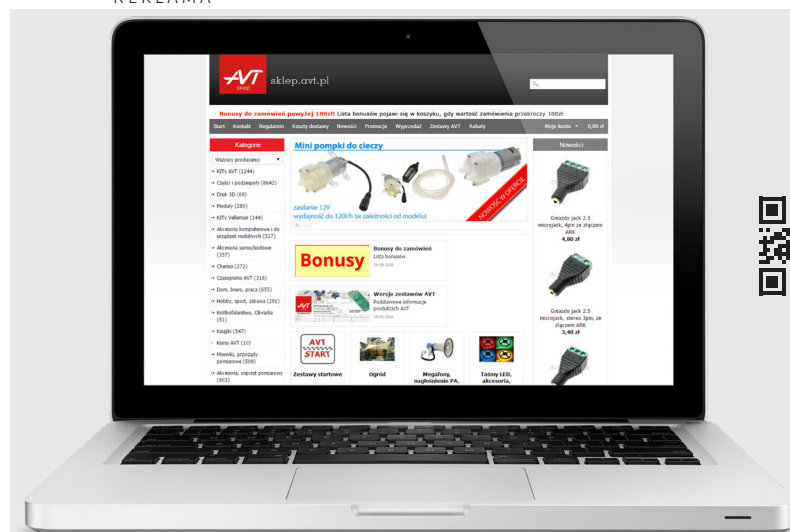
$$\Theta_{sa} = \frac{(150^\circ\text{C} - 25^\circ\text{C}) -$$

$$\frac{17,8 \text{ W} \cdot (1^\circ\text{C/W} + 0,12^\circ\text{C/W})}{17,8 \text{ W}} = 5,9^\circ\text{C/W}$$

Co oczywiste, czym niższy parametr Θ_{sa} radiatora, tym bardziej efektywne odprowadzanie ciepła, a zatem niższa temperatura końcówek mocy. To tyle, jeśli chodzi o podstawowe informacje dotyczące obsługi peryferiów, w związku z czym przejdźmy do szczegółów konstrukcyjnych.

Robert Wołgajew, EP

REKLAMA



<http://sklep.avt.pl>

SKLEP FIRMOWY
(sprzedaż na miejscu,
obsługa zamówień z odbiorem
osobistym):

tel.: 22 257 84 66

Sklep stacjonarny
(ul. Leszcynowa 11, Warszawa – Żerań)
czynny w godzinach:

poniedziałek – piątek: 08:00 – 16:45

(czwartek do 17:45)

sobota: 10:00 – 13:45

