

Przedwzmacniacz audio (2)

Jakość toru sygnałowego jest tak dobra, jak dobry jest jego najłabszy element. Ta zasada dotyczy również elektroakustycznego toru sygnałowego. Entuzjaści dobrego brzmienia wkładają dużo wysiłku i ponoszą spore wydatki na skompletowanie odpowiedniego źródła sygnału, wzmacniacza mocy i zestawów głośnikowych. Ci bardziej „ortodoksyjni” wierzą w zbawienny wpływ materiałów, z których są wykonane kable sygnałowe, głośnikowe, sieciowe i obudowy. Przedwzmacniacz jest elementem toru, który często jest uważany za mniej istotny dla jakości odtwarzanego dźwięku. W obiegowej opinii jest kojarzony tylko z elementem regulującym poziom odtwarzanego sygnału. Takie uproszczone podejście może w konsekwencji powodować degradację sygnału o wiele istotniejszą, niż zastosowanie przetwornika C/A czy wzmacniacza mocy o trochę gorszych parametrach.

Rekomendacje: przedwzmacniacz audio wysokiej klasy, który przyda się do wykonania domowego zestawu audio.

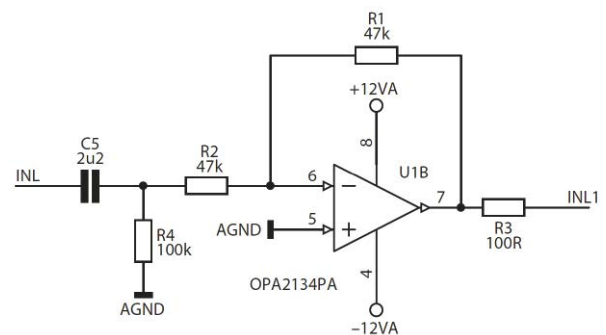
Wzmacniacz wstępny

Wzmacniacz wstępny został zbudowany w topologii wzmacniacza odwracającego – jego schemat ideowy zamieszczono na rysunku 8. Rezystancja wejściowa takiego wzmacniacza jest równa wartości rezystancji R2 (tu – 47 kΩ). Wzmocnienie $G=R1/R2=1$. Użyto tu podwójnego wzmacniacza operacyjnego w obudowie DIL8 (lub SMD na tak zwanym stoliku).

Wybierając układ scalony wzmacniacza, należy zachować ostrożność. Zastosowanie układu o dużym paśmie przenoszenia może skutkować wzbudzeniem się. Pojedyncze, szybkie wzmacniacze umieszczane w obudowach 8-nóżkowych mają wejście do kompensowania pasma przenoszenia (na przykład AD797). Podwójne wzmacniacze takiego wejścia nie mają. W testowanym układzie wzbudzał się układ OPA2406. To podwójny szybki wzmacniacz o paśmie przenoszenia 20 MHz. Nie było natomiast problemów ze wzmacniaczami projektowanymi do układów audio: NE5532 i OPA2134. Dobierając wzmacniacz scalony, należy sprawdzić oscyloskopem, czy się on nie wzbudza. Na wejściu układu zamontowano kondensator C5 odcinający składową stałą. Najlepiej, gdyby to był kondensator foliowy o dobrych parametrach lub elektrolityczny – bipolarny. Ja zastosowałem kondensatory elektrolityczne, bipolarne o pojemności 4,7 μF produkowane przez japońską firmę Nichicon.

Jeżeli mamy pewność, że w torze nie ma i nie będzie składowej stałej w żadnym z doprowadzonych sygnałów, to kondensatory C5 i C6 można zewrzeć. Wydajność prądowa wzmacniacza operacyjnego gwarantuje, że obciążenie wyjścia rezystancją PGS2311 nie spowoduje wzrostu zniekształceń. Wzmacniacz wstępny, chociaż według mnie bardzo wskazany, nie jest niezbędny. Można go pominąć przez zdemontowanie U1 i zrobienie zwór pomiędzy nóżkami 6-7 i 1-2.

Układ U1 jest zasilany napięciem symetrycznym ±12 V. To napięcie dostarczane przez moduł zasilacza jest dodatkowo filtrowane kondensatorami C13 i C15 o pojemności 220 μF oraz blokowane parami



Rysunek 8. Schemat wzmacniacza wstępnego (jeden kanał)

kondensatorów: ceramicznym 100 nF i tantalowym elektrolitycznym 10 μF.

Regulator sygnału audio

Schemat ideowy Regulacja sygnału audio jest wykonywana przez układ PGA2311. Ponieważ działanie tego układu pokrótce omówiono na początku artykułu, zajmijmy się teraz jego aplikacją. Sygnały z wyjścia wzmacniacza wstępnego INL1 i INR1 są podawane na wejście ViL i ViR, a po regulacji wzmocnienia/tłumienia trafiają na wyjście VoL i VoR. Te wyjścia są jednocześnie wyjściami wewnętrznego wzmacniacza operacyjnego i dlatego na wyjściu nie zastosowałem kolejnego wzmacniacza zbudowanego w oparciu na wzmacniaczu operacyjnym,

REKLAMA

Projekty na

STM32

www.stm32.eu

life.augmented

KAMAMI

DODATKOWE MATERIAŁY NA FTP:

<ftp://ep.com.pl>

USER: 77642, PASS: 3220ppmm

W ofercie AVT*

AVT-5542

Podstawowe informacje:

- Zbudowany z użyciem wzmacniacza PGA2311 o cyfrowo regulowanym tłumieniu/wzmocnieniu.
- Sterowany mikrokontrolerem PIC16F1936.
- Złożony z trzech płytek: sterownika, regulatora i zasilacza.
- Wyposażony w czytelny wyświetlacz LCD.
- Możliwość zdalnego sterowania pilotem na poczerwień.
- Zegar czasu rzeczywistego.
- Selektor wejść umożliwia wybór 1 z 3 źródeł.
- Przetwarzanie źródeł za pomocą przełączników.

Projekty pokrewne na FTP:

(wymienione artykuły są w całości dostępne na FTP)

AVT-5382	PRE4562 – przedwzmacniacz liniowy audio (EP 2/2013)
AVT-1670	Stereofoniczny regulator barwy dźwięku (EP 4/2012)
AVT-1634	Przedwzmacniacz z TDA1524A (EP 8/2011)
AVT-566	Procesor audio z wejściem S/PDIF (EP 3-4/2004)
AVT-5082	Cyfrowy procesor dźwięku (EP 9/2002)
AVT-5066	Trójwejściowy przedwzmacniacz Hi-Fi (EP 6/2002)

* Uwaga: Zestawy AVT mogą występować w następujących wersjach:
 AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.
 AVT xxxx A płytką drukowaną PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.
 AVT xxxx A+ płytką drukowaną i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.
 AVT xxxx B płytką drukowaną (lub płytki) oraz komplet elementów wymienionych w załączniku pdf
 AVT xxxx C to nic innego jak zamontowany zestaw B, czyli elementy wlutowane w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku pdf
 AVT xxxx CD oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje, to niezbędne oprogramowanie można ściągnąć, klikając w link umieszczony w opisie kitu)
 Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz (UK, A, A+, B lub C). <http://sklep.avt.pl>

bo to rozbudowałyby układ, a efekt byłby żaden. Część analogowa PGA2311 jest zasilana napięciem ± 5 V. Podobnie jak w przypadku zasilania układu U1, zastosowałem dodatkowe filtrowanie kondensatorami 220 μ F i blokowanie kondensatorami 100 nF i tantalowym 10 μ F.

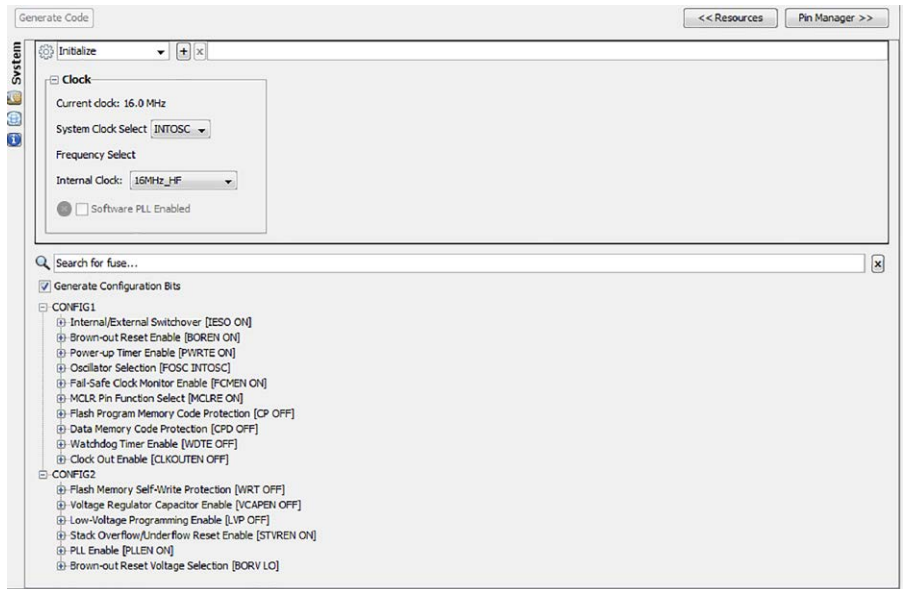
Wejście sterujące ZCEN jest przeznaczone do włączania/wyłączania układu regulacji poziomu w zerze sygnału. Jeżeli ten układ jest włączony, to zmiana cyfrowych nastaw poziomu sygnału wyjściowego odbywa się w momencie, kiedy sygnał wejściowy ma amplitudę równą 0 V (potencjał AGND). Zapobiega to powstawaniu zaburzeń w momencie sterowania poziomem, ale wprowadza dodatkowe opóźnienie sterowania. Takie opóźnienie w naszym urządzeniu zupełnie nie przeszkadza i dlatego regulacja w zerze jest włączona na stałe.

Wejście MUTE służy do sterowania wyciszeniem układu. Wymuszenie logicznego stanu niskiego powoduje sprzętowe wpisanie do rejestrów sterujących wartości 0x00, czyli całkowite wyciszenie układu. U nas ten mechanizm jest wyłączony (poziom logiczny wysoki). Sprzętowo sterowanie wejściem MUTE z racji izolacji galwanicznej mas musiałyby być izolowane transoptorem. Taki sam efekt można osiągnąć przez programowe zapisanie rejestrów sterujących.

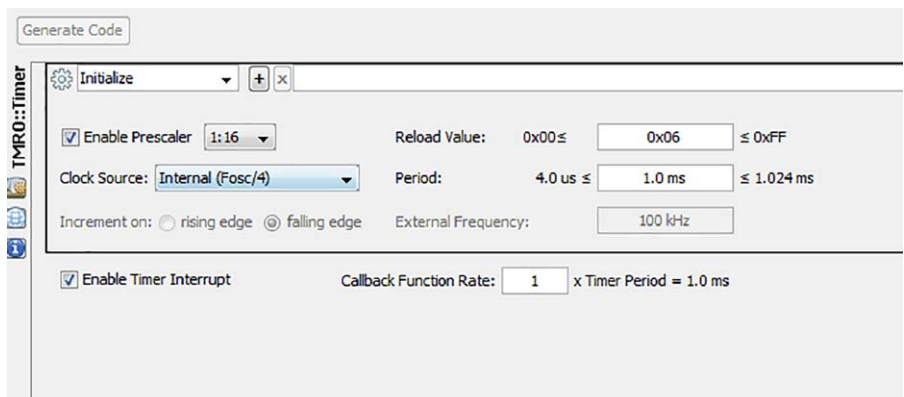
Wyjście sygnału może być zwierane do masy przez przełącznik K5 MUTE. To dodatkowe rozszerzenie układu umożliwia wyeliminowanie stuków generowanych przez układ w momencie włączania zasilania układów analogowych. Sterownik mikroprocesorowy najpierw uaktywia sygnał MUTE i zwiiera poprzez rezystory 100 Ω wyjście PGA2311. Potem włącza zasilanie sekcji analogowej, odczekaże zadany czas i dezaktywuje sygnał MUTE. W czasie włączania zasilania, kiedy na wyjściu PGA2311 mogą się pojawić stany nieustalone, to wyjście jest zwarte. Kiedy układ jest zasilony i wszystko się ustabilizuje, przełącznik K5 rozwiera styki i zwarcie do masy jest usuwane. Tego nie można osiągnąć przez programowe lub sprzętowe sterowanie wyciszeniem w układzie PGA2311.

Sterowanie regulatorem audio

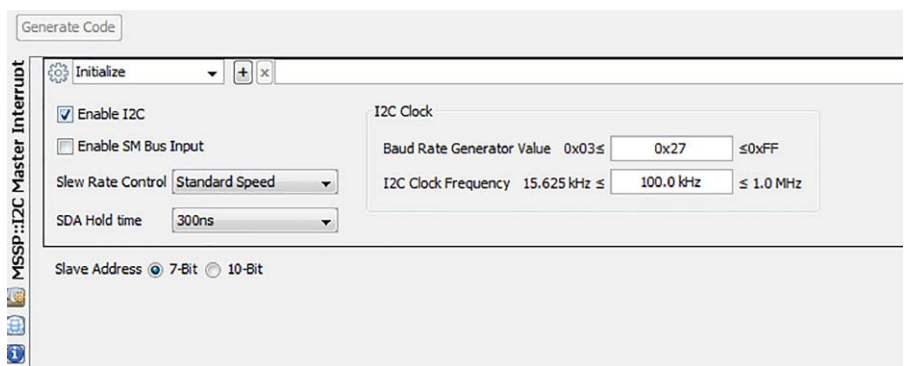
Za sterowanie regulatorem odpowiada program sterujący zapisany w pamięci mikrokontrolera PIC16F1936. Program został napisany w języku C i skompilowany kompilatorem MPLAB XC8. Do napisania, skompilowania, uruchomienia i zapisania w pamięci Flash mikrokontrolera użyłem pakietu MPLAB X IDE. Bardzo użyteczna okazała się również firmowa wtyczka MCC – MPLAB Code Configurator. Użyłem jej do szybkiego skonfigurowania układów peryferyjnych: licznika, układu PWM, portów i interfejsu I²C.



Rysunek 9. Konfiguracja bitów konfiguracyjnych i taktowania



Rysunek 10. Konfiguracja licznika Timer0



Rysunek 11. Inicjalizacja interfejsu I²C: 100 kHz, 7-bitowy adres slave

Każde urządzenie, które jest obsługiwane przez człowieka, musi być wyposażone w interfejs użytkownika. Za jego pomocą można wprowadzać nastawy do urządzenia i odczytywać dane przekazywane przez urządzenie. W tym przedwzmacniaczu interfejs jest zbudowany z: alfanumerycznego wyświetlacza LCD o rozdzielczości 4x20 znaków, impulsatora (enkodera obrotowego) z przyciskiem zintegrowanym z osią oraz odbiornika podczerwieni IR, w komplecie z pilotem zdalnego sterowania pracującym w kodzie RC5.

Interfejs użytkownika jest oparty na programowych procedurach obsługujących

wyświetlacz i jego podświetlenie, enkoder i układ zdalnego sterowania.

Po włączeniu zasilania lub zerowaniu mikrokontrolera w pierwszej kolejności są wykonywane procedury inicjalizacyjne. Najpierw jest wywoływana funkcja *SYSTEM_Initialize()* wygenerowana przez wtyczkę MCC MPLAB Code Configurator. Zastosowałem to bardzo użyteczne narzędzie po to, aby szybko i poprawnie wprowadzić nastawy:

- Bitów konfiguracyjnych mikrokontrolera.
- Układ taktowania.
- Porty wejścia/wyjścia.

- Kontroler przerwań.
- Licznik/timer TMR0 do odliczania opóźnień.
- Układ PWM ECCP, łącznie z licznikiem TMR2.

Bit'y konfiguracyjne ustawiają taktowanie przez wewnętrzny oscylator RC i włączają liczniki opóźnień przy włączeniu zasilania (PWRT=ON). Programowanie niskonapięciowe jest wyłączone (LWP=OFF), wyłączony jest też licznik watchdog (WDTE=OFF).

Mikrokontroler jest taktowany częstotliwością 16 MHz z wewnętrznego oscylatora RC. Okno MCC z ustawieniami taktowania i bitów konfiguracyjnych zostało pokazane na **rysunku 9**. Do odliczania kalibrowanych użyto licznika TMR0. Odlicza on interwały o długości 1 ms i zgłasza przerwanie. Na **rysunku 10** pokazano inicjalizację licznika z poziomu MCC. MCC generuje kod z inicjalizacją licznika (**listing 1**) i po zaznaczeniu opcji *Enable Interrupt* - funkcję obsługi przerwania, która wywołuje procedurę użytkownika *TMR0-CallBack* (**listing 2**). Oprócz odliczania opóźnień co 1 ms jest wywoływana funkcja obsługi enkodera obrotowego i jest sprawdzany warunek wyświetlania czasu z zegara RTC.

Interfejs I²C używany do komunikacji z układem zegara RTC typu DS1307 jest

Listing 1. Inicjalizacja licznika TMR0

```
void TMR0_Initialize(void) {
/* Set TMR0 to the options selected in the User Interface
PSA assigned; PS 1:16; TMRSE Increment_hi_lo; mask the nWPUEEN and INTEDG
bits */
OPTION_REG = (OPTION_REG & 0xC0) | 0xD3 & 0x3F;
// TMR0 6;
TMR0 = 0x06;
// Load the TMR value to reload variable
timer0ReloadVal = 6;
// Clear Interrupt flag before enabling the interrupt
INTCONbits.TMR0IF = 0;
// Enabling TMR0 interrupt
INTCONbits.TMR0IE = 1;
}
```

Listing 2. Funkcja użytkownika wywoływana co 1 ms z obsługi przerwania

```
#define timeout bits.timeout
extern unsigned int Delay;
void TMR0_CallBack(void) {
/* Add your custom callback code here
this code executes every 1 TMR0 periods */
if(!timeout) {
--Delay;
if(Delay==0) timeout=1;
}
//obsługa enkodera
encoder();
CheckRtc(); //sprawdzenie czy wyświetlić czas
}
```

również konfigurowany przez MCC. W tym wypadku jest to więcej niż konfiguracja, bo MCC generuje również kompletne, nieblokujące procedury obsługi magistrali oparte na systemie przerwań. Częstotliwość zegara transmisji SCL ustalono na 100 kHz

(**rysunek 11**). Na **listingu 3** zamieszczono funkcje inicjalizacji interfejsu.

Przebieg PWM użyty do sterowania podświetleniem wyświetlacza jest generowany przez sprzętowy moduł ECCP1, współpracujący z licznikiem Timer2. Początkowo

REKLAMA

► POLECANY PRODUKT

ALU-29

Preparat do lutowania aluminium

- Do lutowania aluminium i jego stopów.
- Wystarczy kropla preparatu, cyna i zwykła lutownica.
- Daje spoinę odporną mechanicznie i zapewnia dobry kontakt elektryczny lutowanych metali.
- Do lutowania ekranów, uziomów radiatorów, baterii akumulatorów Li-Po, Li-Ion, naprawiania chłodziń i inne zastosowania.



STS-10

Preparat do lutowania stali nierdzewnej, powierzchni chromowanych i innych

- Do lutowania stali nierdzewnej, stali chromowanej, stali ocynkowanej i innych metali, które trudno zlutować w warunkach warsztatu elektronika.
- Wystarczy kropla preparatu, cyna i zwykła lutownica.
- Daje spoinę odporną mechanicznie i zapewnia dobry kontakt elektryczny lutowanych metali.
- Do lutowania wyprowadzeń baterii, doprowadzeń masy obudów układów scalonych i inne zastosowania.



Uwaga!
20% rabatu

dla Czytelników EP do 30 lipca 2016. Wystarczy przy zakupie podać hasło "Elektronika Praktyczna". Produkt można zakupić w siedzibie firmy Iviter w Warszawie przy ul. Dzielnej 7a, dzwoniąc pod numer telefonu 22 297 70 30 lub 22 297 70 40 oraz przez internet www.ivichem.pl/sklep

Producent: Iviter, ul. Dzielna 7a, 00-154 Warszawa, tel. 22 297 70 30, e-mail: info@iviter.pl, sklep internetowy: [ivichem http://ivichem.pl/sklep](http://ivichem.pl/sklep)

chciałem zrealizować przebieg PWM programowo z wykorzystaniem przerwań od licznika TMR2. Jednak dla tego mikrokontrolera byłoby to za duże obciążenie. Jak wykazały próby, procedura generująca PWM przeszkadzała procedurze przerwania od TMR0 i przez to źle działała obsługa enkodera i wyświetlacza.

Moduł ECCP jest rozbudowany. Ma możliwość generowania 2 przebiegów PWM do sterowania mostkiem. Nam będzie potrzebny zwykły przebieg PWM z możliwością sterowania wypełnieniem. Skonfigurujemy ECCP1, aby jego częstotliwość wynosiła ok 16,6 kHz (rysunek 12). Ta częstotliwość jest równa częstotliwości przepelniania się licznika TMR2.

Konfigurator MCC generuje standardowo procedurę inicjalizacyjną i przydatne funkcje, między innymi *EPWM1_LoadDutyValue()* ustalającą współczynnik wypełnienia – pokazano ją na listingu 4. Na listingu 5 zamieszczono funkcje *System_Initialize()* i *OSCILLATOR_Initialize()*.

Po inicjalizacji taktowania i używanych modułów peryferyjnych program wywołuje funkcję *BOARDInit()*. Ustala ona wartości początkowe, takie jak: poziom sygnału, aktywne wejście, jasność podświetlenia. Odczytuje też z pamięci EEPROM wcześniejsze zaprogramowane kody RC5. Te wartości są w trakcie obsługiwanego urządzenia zapisywane do pamięci EEPROM i w czasie inicjalizacji odtwarzane. Pokazana na listingu 6 funkcja *BOARDInit()* inicjuje również działanie wyświetlacza LCD, zegara RTC oraz włącza zasilanie układów analogowych modułu regulatora.

Obsługa wyświetlacza LCD

Jak wspomniano, zastosowano wyświetlacz alfanumeryczny mieszczący 4 linie po 20 znaków. Sterownik ST7066U wyświetlacza różni się od uznawanego za standard HD44780 i wymaga nieco innej inicjalizacji. Magistrala sterująca, organizacja rejestrów sterujących i pamięci obrazu jest identyczna. Kiedy użyłem swoich wielokrotnie sprawdzonych procedur inicjalizacji, to niestety wyświetlacz nie chciał działać. Dla ułatwienia samodzielnego zastosowania tego wyświetlacza, wykonaną i sprawdzoną przeze mnie procedurę inicjalizacyjną (4-bitowa szyna danych) zamieściłem na listingu 7.

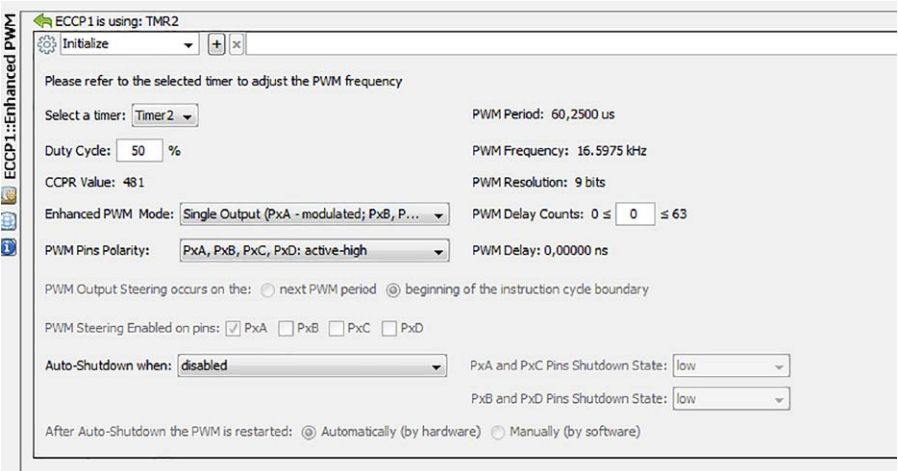
Zegar RTC DS1307

Zegar RTC jest taktowany typowym oscylatorem kwarcowym, „zegarkowym” o częstotliwości 32768 Hz. Oscylator jest dołączony do wyprowadzeń X1 i X2 bez kondensatorów obciążających, ponieważ te są zamontowane w strukturze układu. Na wszelki wypadek, na płytce przewidziano miejsce na kondensatory, ale w prototypie nie są one wlutowane. Układ DS1307 jest

Listing 3. Inicjalizacja interfejsu I²C

```
void I2C_Initialize(void) {
    i2c_object.pTrHead = i2c_tr_queue;
    i2c_object.pTrTail = i2c_tr_queue;
    i2c_object.trStatus.s.empty = true;
    i2c_object.trStatus.s.full = false;
    i2c_object.i2cErrors = 0;

    // BF RCinprocess_TXcomplete; UA dontupdate; SMP Standard Speed; P stopbit_
    // notdetected
    //S startbit_notdetected; R_nW write_noTX; CKE Idle to Active; D_nA lastbyte_
    // address;
    SSPSTAT = 0x80;
    // SSPM0 FOSC/4_SSPxADD; SSPEN enabled; WCOL no_collision; SSPOV no_overflow;
    //CKP Idle:Low, Active:High;
    SSPCON1 = 0x28;
    // BOEN disabled; AHEN disabled; SBCDE disabled; SDAHT 300ns; DHEN disabled;
    //ACKTIM ackseq; PCIE disabled; SCIE disabled;
    SSPCON3 = 0x08;
    // Baud Rate Generator Value: SSPADD 39;
    SSPADD = 0x27;
    /* Byte sent or received */
    // clear the master interrupt flag
    PIRbits.SSPIF = 0;
    // enable the master interrupt
    PIE1bits.SSPIE = 1;
}
```



Rysunek 12. Konfiguracja ECCP1

typowym zegarem z kalendarzem. Czas jest odmierzany z dokładnością do 1 sekundy. Dostęp do zegara i kalendarza odbywa się przez interfejs I²C. Wykaz rejestrów układu DS1307 umieszczono w tabeli 2.

Funkcja *InitRtc()* (listing 8) zapisuje do rejestru konfiguracyjnego wartość 0x10: SQWE=1i RS2:RS1=00. Na wyjściu SQW/OUT jest generowany przebieg o częstotliwości 1 Hz i wypełnieniu 50%. Mikrokontroler testuje stan tego wyjścia w obsłudze przerwania od TMR0 co 1 ms (listing 9) i jeżeli wykryje zbocze opadające, to odczytuje zawartość rejestrów zegara. Wykrycie zmiany licznika minut powoduje wywołanie procedury wyświetlenia godziny i daty.

Po wykonaniu wszystkich czynności inicjalizacyjnych program wyświetla ekran główny i przechodzi do pętli głównej. W pętli głównej można wykonać następujące działania:

- Regulować poziom sygnału w całym zakresie od -95 do +32 dB z krokiem co 1 dB. Regulacja odbywa się poprzez obracanie osi enkodera obrotowego lub przez przyciskanie klawiszy „w górę” i „w dół” na pilocie zdalnego sterowania.

Sterowanie za pomocą pilota musi być poprzedzone zaprogramowaniem kodów RC5 przez wywołanie funkcji programującej kody RC5.

- Zmieniać aktywne wejście. Zmiana może się odbywać po przyciśnięciu odpowiedniego przycisku (po zaprogramowaniu kodów) lub przez wejście do menu funkcyjnego i wywołanie funkcji zmiany aktywnego wejścia.
- Uaktywnić funkcję wyciszania *Mute* – tylko za pomocą zaprogramowanego wcześniej przycisku pilota.
- Wyłączyć całą analogową część regulatora (*Power Off*) – tylko za pomocą zaprogramowanego wcześniej przycisku pilota.
- Wejść do menu funkcyjnego po przyciśnięciu oski enkodera obrotowego.

Pętlę menu głównego pokazano na listingu 10. Na początku pętli jest wykonywane sprawdzenie, czy nie została obrócona lub przyciśnięta oś enkodera i czy na linii dołączonej do wyjścia odbiornika podczerwieni nie wykryto poziomu niskiego (początek nadawania kodu RC5). Jeżeli

```

Listing 4. Inicjalizacja ECCP1 i funkcja EPWM_LoadDutyValue
void EPWM1_Initialize(void) {
/* Set the PWM to the options selected in MPLAB® Code Configurator
P1M single; CCP1M P1AP1Chi_P1BP1Dhi; DC1B 16; */
    CCP1CON = 0x1C;
// CCP1AS0 disabled; PSS1AC0 P1AP1C_0; CCP1ASE operating; PSS1BD0 P1BP1D_0;
    ECCP1AS = 0x00;
// P1DC0 0; P1RSEN automatic_restart;
    PWMLCON = 0x80;
// STR1A P1A_to_CCP1M; STR1C P1C_to_port; STR1B P1B_to_port;
// STR1D P1D_to_port; STR1SYNC start_at_begin;
    PSTR1CON = 0x01;
// CCPR1L 120;
    CCPR1L = 0x78;
// CCPR1H 0x0;
    CCPR1H = 0x00;
// Selecting Timer2
    CCP1TMR0bits.C1TSEL = 0x0;
}

void EPWM1_LoadDutyValue(uint16_t dutyValue) {
// Writing to 8 MSBs of pwm duty cycle in CCPRL register
    CCPR1L = ((dutyValue & 0x03FC) >> 2);
// Writing to 2 LSBs of pwm duty cycle in CCPCON register
    CCP1CON = (CCP1CON & 0xCF) | ((dutyValue & 0x0003) << 4);
}

void TMR2_Initialize(void) {
/* Set TMR2 to the options selected in the User Interface
TMR2ON off; T2CKPS 1:1; T2OUTPS 1:1; */
    T2CON = 0x00;
// PR2 240;
    PR2 = 0xF0;
// TMR2 0x0;
    TMR2 = 0x00;
// Clearing IF flag.
    PIR1bits.TMR2IF = 0;
// Start TMR2
    TMR2_StartTimer();
}
    
```

```

Listing 5. Funkcja SYSTEM_Initialize i OSCILLATOR_Initialize
void SYSTEM_Initialize(void) {
    OSCILLATOR_Initialize();
    PIN_MANAGER_Initialize();
    TMR0_Initialize();
    I2C_Initialize();
    TMR2_Initialize();
    EPWM1_Initialize();
}

void OSCILLATOR_Initialize(void) {
// SPLLEN disabled; SCS INTOSC; IRCF 16MHz_HF;
    OSCCON = 0x7A;
// OSTs intosc; HFIOFR disabled; HFIOFS not0.5percent_acc; PLLR disabled;
T10SCR disabled; MFIOFR disabled; HFIOFL not2percent_acc; LFIOFR disabled;
    OSCSTAT = 0x00;
// TUN 0x0;
    OSCTUNE = 0x00;
// Set the secondary oscillator
}
    
```

żaden z tych warunków nie został spełniony, to jest tylko wywoływana funkcja *DispCzas()* sprawdzająca, czy trzeba wyświetlić godzinę (zmienił się stan licznika minut od ostatniego wyświetlenia) i ewentualnie ją wyświetla. Wyświetlanie godziny jest też uwarunkowane ustawieniem w menu. Można tak skonfigurować sterownik, aby godzina nie była w ogóle wyświetlana.

Obrócenie ośki w prawo lub odebranie kodu równego wartości stałej *RC5_UP* powoduje zmniejszenie tłumienia o 1 dB, wyświetlenie nowej wartości na ekranie wyświetlacza i ewentualnie zmodyfikowanie linijki obrazującej wartość poziomu sygnału. Po osiągnięciu wartości +35 dB dalsze zwiększanie wzmocnienia nie jest możliwe. Bardzo podobnie działa zwiększenie tłumienia po obrocie ośki w lewo lub odebraniu kodu o wartości *RC5_DWN*. Po osiągnięciu tłumienia -95 dB dalsze zwiększanie tłumienia nie jest wykonywane. Podczas regulowania poziomu sygnału audio nastawy są zapisywane do PGA2311 poprzez magistralę SPI. Każda sekwencja zapisywania rozpoczyna się wyzerowaniem linii CS. Jeden cykl zapisu polega na przesłaniu słowa 16-bitowego: pierwsze 8 bitów określa poziom sygnału w kanale prawym, a kolejne 8 bitów poziom sygnału w kanale lewym. Bity są przesyłane od najstarszego w czasie narastającego zbocza zegara (rysunek 13). Po przesłaniu 16 bitów linia CS musi być ustawiona. Interfejs SPI transmitujący słowa 16-bitowe zrealizowano programowo (listing 11).

Funkcja *SendPGA* ma dwa 8-bitowe argumenty *volL* i *volR* określające ustawiany poziom sygnału. Z tych dwóch 8-bitowych wartości jest tworzona zmienna 16-bitowa *vol* i jej zawartość jest wysyłana przez SPI bit po bicie. Jak wiemy, transmisja odbywa się poprzez barierę galwaniczną zrealizowaną na transoptorach. W czasie przełączania w transoptorze

Tabela 2. Rejestry układu DS1307

adres	B7	B6	B5	B4	B3	B2	B1	B0	Funkcja	zakres
00h	CH	Dziesiątki sekund			Sekundy			Sekundy	00-59	
01h	0	Dziesiątki minut			Minuty			Minuty	00-59	
02h		24	Dziesiątki godzin		Godziny			Godziny	0-23	
		12	PM/AM	Dziesiątki godzin	1-12 AM/PM					
03h	0	0	0	0	0	dni		Dni tygodnia	01-07	
04h	0	0	Dziesiątki dni mies.		Dni miesiąca			Dni miesiąca	01-31	
05h	0	0	0	Dziesiątki miesiąca		miesiące			Miesiące	01-12
06h	Dziesiątki roku			jednostki roku			Rok	00-99		
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Rejestr sterujący	
08h-3fh									RAM 56 bajtów	00h-fff

CH – clock halt CH=1 oscylator zatrzymany
 OUT – stan wyjścia SQW/OUT, kiedy generowanie przebiegu cyfrowego na tym wyjściu jest wyłączone (SQWE=0)
 SQWE – włączenie/wyłączenie przebiegu cyfrowego na wyjściu SQW/OUT
 RS2:RS1 – częstotliwość na wyjściu SQW/OUT

przeznaczonym do zastosowań uniwersalnych „dodawane” są opóźnienia rzędu dziesiątek mikrosekund. Dla naszego transoptora TLP281 są to czasy $T_{on}=2 \mu s$, $t_s=25 \mu s$ i $T_{off}=40 \mu s$ (rysunek 14). Z tego powodu częstotliwość przełączania poziomów logicznych na wyjściu transoptora jest ograniczona.

Jeżeli zsumujemy te czasy, to otrzymamy $67 \mu s$ i maksymalna częstotliwość sygnału to ok. 15 kHz. W czasie uruchamiania urządzenia dołączyłem 2-kanalowy oscyloskop – jeden kanał do przebiegu zegarowego SCK na wyjściu portu mikrokontrolera i drugi do wyjścia transoptora SCKI. Do cyklicznie wywoływanej procedury `SendPGA()` dodałem programowe opóźnienia `del()`. Tak dobrałem jego czas, aby sygnał na wyjściu z kolektora tranzystora transoptora nie był zniekształcony w porównaniu z przebiegiem źródłowym.

Przyśnięcie oski enkodera w pętli głównej wywołuje menu funkcyjne sterownika. Jego obsługę zaimplementowano w pokazanej na **listingu 12** funkcji `MenuFun()`. Menu jest wielopoziomowe i pozwala na:

- Zmianę aktywnego wejścia selektora sygnału audio.
- Ustawienie zegara i kalendarza.
- Zaprogramowanie kodów pilota RC5.
- Ustawienie podświetlania wyświetlacza LCD i włączenie/wyłączenie wyświetlacza LCD i godziny i daty na ekranie głównym.

Funkcję przeznaczoną do wykonania wybiera się przez kręcenie osi enkodera obrotowego. Wybraną funkcję wskazuje strzałka z lewej strony ekranu LCD.

Jedną z ważniejszych funkcji z punktu widzenia możliwości obsługi jest programowanie kodów pilota RC5. W trakcie programowania są sekwencyjnie wybierane funkcje wykonywane przez przycisk na pilocie. Po naciśnięciu przycisku kod jest odbierany przez odbiornik, wyświetlany na ekranie i zapamiętywany w pamięci EEPROM mikrokontrolera (**listing 13**).

Montaż i uruchomienie

Przedwzmacniacz jest zbudowany z trzech modułów umieszczonych na osobnych płytkach: płytce sterownika, płytce regulatora i płytce zasilacza. Wszystkie płytki mają identyczne wymiary i są przeznaczone do zamontowania w obudowie o wymiarach $165 \text{ mm} \times 85 \text{ mm} \times 125 \text{ mm}$.

Płytkę sterownika Schemat montażowy płytki sterownika pokazano na **rysunku 15**. Jej montaż rozpoczynamy od przyłutowania elementów po stronie elementów. Montujemy tu wszystkie elementy oprócz wyświetlacza i enkodera obrotowego. Do wyprowadzeń wyświetlacza trzeba przyłutować przyciętą na długość listwę goldpinów o rastrze 2,54 mm.

Listing 6. Inicjalizacja nastaw sterownika

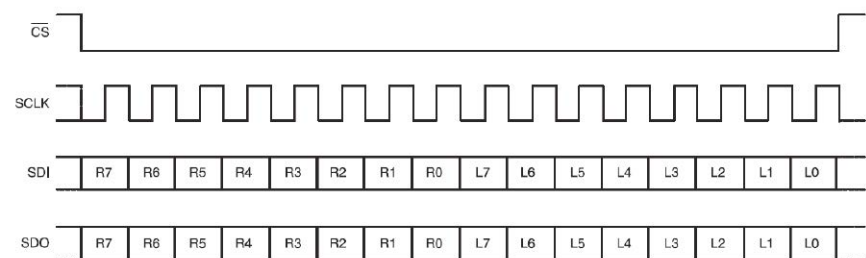
```

/*****
Inicjalizacja działania przedwzmacniacza po włączeniu zasilania
*****/
void BOARDInit(void)
{
    unsigned char vol, vols;
    //sekwencja zalaczenia zasilania
    STBY=0;asm(„nop”);asm(„nop”);asm(„nop”);asm(„nop”); //moduł analogowy
    wylaczony
    MUTE=1; //uklad MUTE dziala
    Delay_ms(100);
    STBY=1; //uklad analogowy wylaczony
    Delay_ms(500);
    MUTE=0; //wylacz MUTE
    SCK=0;SDO=0;CS=1;//wartosci poczatkowe SPI dla PGA2311
    OPTION_REGbits.nWPUN=0;
    PEIE=1; //odblokowanie przerwań od peryferii
    ei(); //odblokowanie wszystkich przerwań
    Delay_ms(500);
    LcdInit(); //inicjalizacja wywietlacza
    InitRtc(); //inicjalizacja RTC
    //ta sekwencja wykonuje sie po pierwszym zaprogramowaniu pamieci FLASH
    if(DATAEE_ReadByte(ADR_INIT)!=0x29){
        InitCzas();
        DATAEE_WriteByte(ADR_VOL,20);//wartosc tlumienia/wzmocnienia
        DATAEE_WriteByte(ADR_FUN,0);//numer glownej funkcji
        DATAEE_WriteByte(ADR_FUN1,0);//numer glownej funkcji
        DATAEE_WriteByte(ADR_CWYS,0);//nie wywietlaj czasu na ekranie
        DATAEE_WriteByte(ADR_JW,5);//jasnosc wywietlacza
        DATAEE_WriteByte(ADR_INP,0);//pierwsze wejście aktywne
        DATAEE_WriteByte(ADR_RC5_ZN,0);//znacznik zaprogramowania kodów - nie
        zaprogramowane
        DATAEE_WriteByte(ADR_INIT, 0x29);
    }
    EPWM1_LoadDutyValue(DATAEE_ReadByte(ADR_JW)*100);//odtwórz jasność
    wywietlacza
    if(DATAEE_ReadByte(ADR_RC5_ZN)==1) GetRC5Code(); //jezeli kody sa
    w pamieci eeprom, to odczytaj i wpisz do RAM
    vol=DATAEE_ReadByte(ADR_VOL);
    vols=(vol*2-1);
    SendPGA(vols,vols); //odtwórz poziom sygnału
    SetIn(DATAEE_ReadByte(ADR_INP)); //ustawienie aktywnego wejścia
}
    
```

Listing 7. Inicjalizacja sterownika ST7066U

```

/*****
Inicjalizacja sterownika LCD
*****/
void LcdInit(void){
    Delay_ms(100);
    RS=0;
    asm(„nop”);asm(„nop”);asm(„nop”);asm(„nop”);
    asm(„nop”);asm(„nop”);asm(„nop”);asm(„nop”);
    EN=0;
    asm(„nop”);asm(„nop”);asm(„nop”);asm(„nop”);
    asm(„nop”);asm(„nop”);asm(„nop”);asm(„nop”);
    WriInit(2);
    Delay_ms(5);
    WriInit(2);
    Delay_ms(5);
    WriInit(8);
    Delay_ms(5);
    WriteRi(0x0f);
    Delay_ms(5);
    WriteRi(8);
    Delay_ms(5);
    WriteRi(1);
    Delay_ms(5);
    WriteRi(6);
    Delay_ms(5);
    WriteRi(0x0c);
    Delay_ms(5);
}
    
```



Rysunek 13. Przesłanie danych do PGA 2311

Listing 8. Inicjalizacja DS1307

```

/*****
Wykrywanie zbocza opadającego sygnału RTC
*****/
void CheckRtc(void) {
    if(rtc==0) {
        if(rtc0==1&& RTC==1)
        {
            rtc0=0;rtc1=1;
        }
        if(rtc1==1&&RTC==0)
        {
            rtc=1;rtc1=0;rtc=1;
        }
    }
}
    
```

Potem wyświetlacz trzeba przykręcić na tulejkach dystansowych, przylutować listwę do wyprowadzeń od umownej strony lutowanie (solder side) płytki drukowanej sterownika, tak jak to zostało pokazane na **fotografii 16**. Od tej samej strony lutujemy enkoder obrotowy. W przednim panelu obudowy wycinamy otwór na wyświetlacz i otwór na oś enkodera i za pomocą wkrętów mocujemy

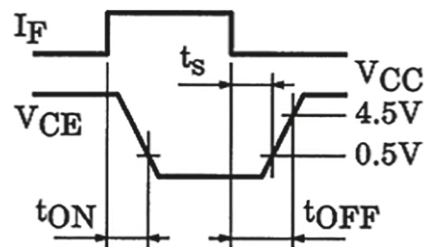
do panelu przedniego. Z lewej strony panelu trzeba wywiercić otwór na soczewkę odbiornika IR. Widok przedniego panelu został pokazany na **fotografii 17**.

Płytką zasilacza Schemat montażowy płytki zasilacza zamieszczono na **rysunku 18**. Płytkę zasilacza po zamontowaniu przykręciłem do spodu obudowy za pomocą wkrętów i tulejek dystansowych. Pomiędzy płytką a metalowym spodem umieściłem na dystansach starą płytkę o takich samych wymiarach, jak płytką zasilacza. Ta płytkę ma za zadanie dystansować kable biegnące pod zasilaczem od fragmentów płytki, przez które jest przesyłane napięcie 230 V AC. Dotknięcie kabla sygnałowego do punktu lutowniczego z tym napięciem i przebięcie izolacji kabla może skutkować porażeniem. W takim rozwiązaniu nie ma

Listing 9. Testowanie stanu linii RTC - wywoływana co 1 ms w obsłudze przerwania od TMR0

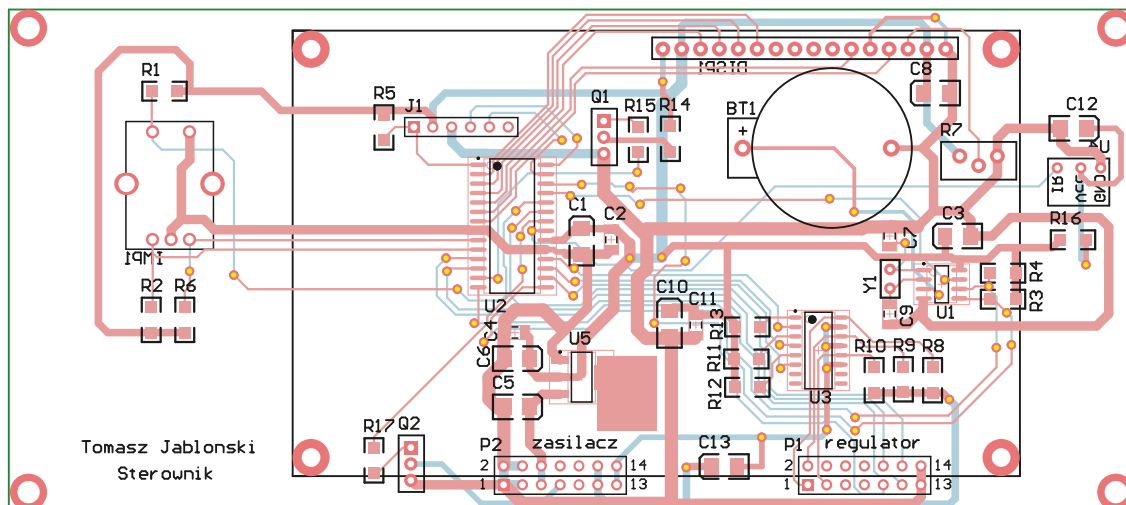
```

/*****
Inicjalizacja zegara RTC
*****/
void InitRtc(void) {
    KonfRtc(0x10);
}
    
```



Rysunek 14. Opóźnienia w czasie przełączania transoptora TLP281

możliwości, aby kabel dotknął wrażliwego miejsca na płytce drukowanej zasilacza. Przy umieszczaniu płytki zasilacza we własnych obudowach wykonanych



Rysunek 15. Schemat montażowy płytki sterownika

REKLAMA

► POLECANY PRODUKT

Platforma handlowa dla firm
www.elektronicspray.com

Cleanser PCC 15

to innowacyjny produkt do czyszczenia płytek drukowanych, a w szczególności pozostałości po lutowaniu. Preparat nie wchodzi w reakcje z materiałami używanymi w elektronice. Szybko wysycha, nie pozostawia żadnych osadów. Przeciwdziała powstawaniu powierzchniowych prądów błądzących. Specjalna szczotka i kapilarna rurka umożliwiają precyzyjne czyszczenie powierzchni.

Cleanser IPA Plus

to czysty, 100-procentowy alkohol izopropylowy. Jest to profesjonalny, wszechstronny środek czyszczący, neutralny chemicznie, szybko odparowuje, nie pozostawia śladów. Specjalna szczotka oraz kapilarna rurka umożliwiają precyzyjne czyszczenie powierzchni. Spray ma zawór działający w kilku położeniach, m.in. w pozycji odwróconej. Preparat IPA można dozować na 3 różne sposoby.

Producent specjalistycznej chemii dla elektroniki

z metalu trzeba o tym pamiętać. Sposób zamontowania płytki zasilacza w obudowie z zamontowanym panelem przednim został pokazany na **fotografii 19**.

Płytką regulatora Montaż płytki regulatora zaczynamy od zamontowania wszystkich elementów oprócz gniazd cinch (schemat montażowy na **rysunku 20**). Do zamontowania wzmacniacza operacyjnego U1 i regulatora U2 można zastosować precyzyjne podstawki. Płytkę regulatora została przewidziana do montażu na tylnym panelu obudowy. Złącza cinch są stojące, do wlutowania w druk. Montuje się je po umownej stronie lutowania, jak wyświetlacz i enkoder obrotowy w płycie sterownika (**fotografia 21**). W tylnym panelu trzeba wywiercić otwory o średnicy ok. 11 mm i przymocować płytkę na tulejkach dystansowych. Widok panelu tylnego pokazano na **fotografiach 22 i 23**.

Uruchomienie układu

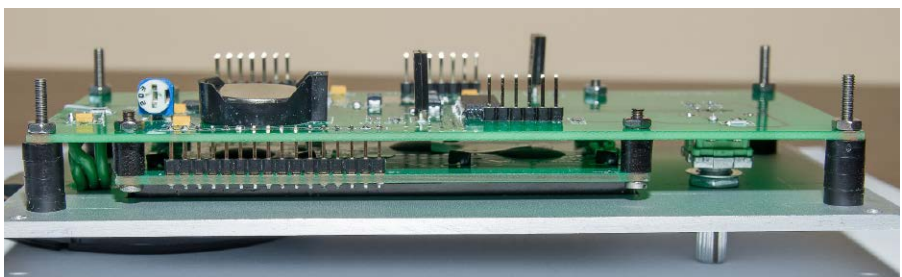
Uruchamianie zmontowanego układu rozpoczynamy od sprawdzenia zasilacza. Transformator T1 ma włączane napięcie pierwotne przez styki przełącznika K1 złączanego z modułu sterownika. Aby można było sprawdzić poprawność działania zasilacza sekcji regulatora, trzeba przyłączyć napięcie 230 V AC do złącza P7. Następnie sprawdzamy poprawność napięć ± 5 V, ± 12 V oraz $+5$ V ($+5$ VDA). Sprawdzony zasilacz można przyłączyć do modułu regulatora złączami P4 (zasilacz) i P12 (regulator) za pomocą taśmy. Przy wyjętych układach U1 i U2 w regulatorze można sprawdzić poprawność podłączenia napięć zasilających.

Jeżeli wszystko jest w porządku, sprawdzamy poprawność napięcia +VDD. Napięcie 230 V AC doprowadzamy do złącza P6 zasilacza. Przy zastosowanym transformatorze, bez obciążenia to napięcie wynosi ok. $+11$ V. Następnie łączymy kablem zasilacz ze sterownikiem: złącze P5 (zasilacz) i P2 (sterownik), i sprawdzamy poprawność napięcia $+5$ V zasilającego układy cyfrowe.

Prawidłowo zasilony mikrokontroler można zaprogramować. Do tego celu przewidziano złącze J1 o wyprowadzeniach zgodnych z wyprowadzeniami programatora Pic-Kit-3. Po zaprogramowaniu mikrokontrolera i połączeniu wszystkich modułów kablami zasilania i sterowania, można uruchomić tor analogowy. Ja do tego celu użyłem starego generatora G430 produkowanego kiedyś przez firmę Meratronik i oscyloskopu 2-kanalowego. Po podaniu na wybrane selektorem wejście sygnału sinusoidalnego o częstotliwości 1 kHz można obserwować regulowany sygnał na wyjściu P15 i P14. Obserwacja tego sygnału pozwala na określenie, czy tor się nie wzbudza i czy w sygnale wyjściowym nie ma innych zaburzeń lub zniekształceń. Trzeba jednak wiedzieć, że ten

Listing 10. Pętla główna programu

```
while(1){
    kod=GetEncoder();//testuj czy obrót oski, przyciśnięcie oski, stan niski
na IR
    if(kod==KOD_IR)
    {
        kod=GetRc5();//odbór kodu IR
        Delay_ms(100);
    }
//wciśnięcie oski enkodera
    if(kod==KOD_IMP_ST)
    {
        while(ST==0);
        MenuFun();//wejście do menu funkcyjnego
        sdata=1;
    }
//regulacja głośniej
    if(kod==KOD_IMP_UP||kod==RC5_UP)
    {
        ++vol;
        if(vol==127) vol=126;
        PosLcd(3,1);
        DispB(vol);
        vols=(vol*2-1);
        SendPGA(vols,vols);
        PosLcd(11,1);
        if(vol>95) DispLinVol(8);
        else DispLinVol(vol/11);
        DATAEE_WriteByte(ADR_VOL, vol);
    }
//regulacja ciszej
    if(kod==KOD_IMP_DWN||kod==RC5_DOWN)
    {
        --vol;
        if(vol==0) vol=1;
        PosLcd(3,1);
        DispB(vol);
        vols=(vol*2-1);
        SendPGA(vols,vols);
        PosLcd(11,1);
        if(vol>95) DispLinVol(8);
        else DispLinVol(vol/11);
        DATAEE_WriteByte(ADR_VOL, vol);
    }
//sterowanie wejściami impulsatora
    if(kod==RC5_IN1)
    {
        PosLcd(3,3);
        DispIn(0);
        SetIn(0);
        DATAEE_WriteByte(ADR_INP,0);
    }
    if(kod==RC5_IN2)
    {
        PosLcd(3,3);
        DispIn(1);
        SetIn(1);
        DATAEE_WriteByte(ADR_INP,1);
    }
    if(kod==RC5_IN3)
    {
        PosLcd(3,3);
        DispIn(2);
        SetIn(2);
        DATAEE_WriteByte(ADR_INP,2);
    }
//komenda MUTE
    if(kod==RC5_MUTE) MuteVol();
//komenda włącz/wyłącz
    if(kod==RC5_POWER) Power();
    }
    DispCzas(1,4);
}
```



Fotografia 16. Montaż wyświetlacza i mocowanie płytki do panelu przedniego obudowy



Fotografia 17. Obudowa z przednim panelem



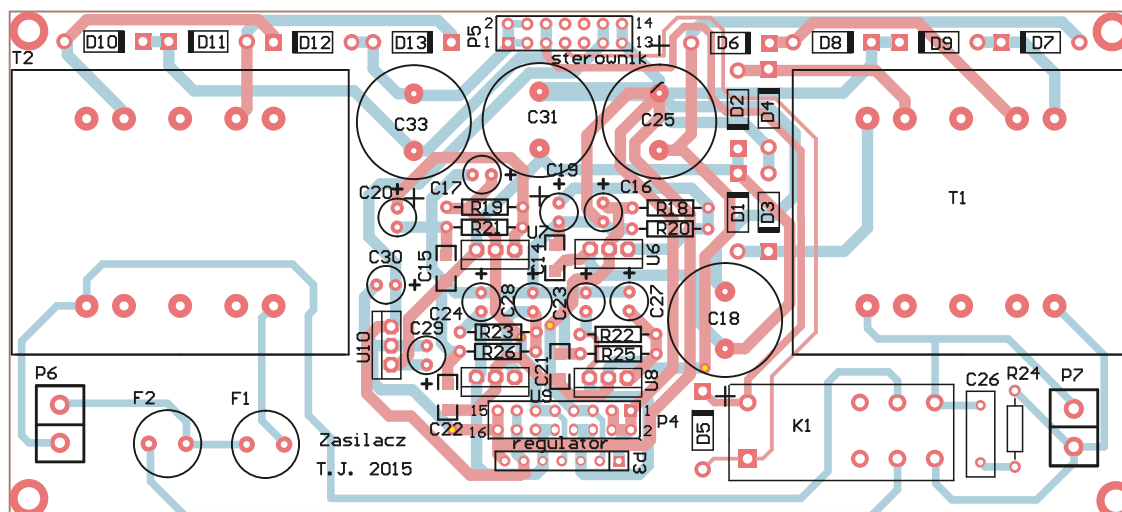
Fotografia 19. Zamontowanie płytki zasilacza w obudowie z panelem przednim

pomiar nie służy do szacowania zniekształceń, ale do określenia, czy w sygnale nie występują duże zaburzenia i czy tor audio pracuje poprawnie. Jeżeli ktoś chciałby wykonać pomiar zniekształceń nieliniowych, to musiałby użyć specjalnych przyrządów pomiarowych.

Konfiguracja i użytkowanie

Po włączeniu zasilania sterownik załącza zasilanie części analogowej i wyświetla ekran główny pokazany na rysunku 24. Umieszczone są na nim: wartość tłumienia/wzmocnienia toru, linijka określająca zgrubnie wartość tłumienia

i aktywne wejście. W tym momencie nie mamy zaprogramowanych kodów pilota i obsługa może się odbywać tylko za pomocą enkodera obrotowego. Obracanie osi powoduje regulowanie poziomu sygnału i wyświetlanie zmodyfikowanej wartości. Po naciśnięciu osi enkodera program wchodzi do menu funkcyjnego



Rysunek 18. Schemat montażowy płytki zasilacza

REKLAMA

► POLECANY PRODUKT

Rozwiązania firmy ACC Silicones w zakresie hermetyzacji, klejenia oraz odprowadzania ciepła w elektronice i energetyce

Lakiery: elektroizolacyjne, akrylowe, silikonowe, maski zrywalne, preparaty do czyszczenia płytek drukowanych.

Zalewy i żele silikonowe: do hermetyzacji elektroniki, przezroczyste odporne na UV do zabezpieczania modułów LED, zalewy i żele termoprzewodzące.

Preparaty termoprzewodzące dozowane: kleje silikonowe 1- i 2-składnikowe, do 3,9 W/mK, pasty termoprzewodzące do 3 W/mK, zalewy i żele silikonowe termoprzewodzące, do 1,55 W/mK (temperatura maksymalna do +275°C), Gap Fillery 2-składnikowe zalewy utwardzalne do 3 W/mK.

Kleje silikonowe dla elektroniki i przemysłu:

1- i 2-składnikowe, neutralne dla elektroniki,
1- i 2-składnikowe do zastosowań przemysłowych.



BL
elektronik

BLElektronik, ul. Rydla 57/17, 30-122 Kraków, tel./faks: 12 394 58 66, tel. 696 483 020, www.blelektronik.com.pl
Sklep internetowy: www.sklep.blelektronik.com.pl

(rysunek 25). Funkcję przeznaczoną do wykonania zaznacza się przez obracanie osią. Przesuwa się wtedy znacznik w postaci strzałki z lewej strony ekranu. Wskazana funkcja jest wykonywana po naciśnięciu osi.

Pierwsza funkcja „Zmień wejście” pozwala na zmianę aktywnego wejścia selektora. Wejścia są zmieniane sekwencyjnie w czasie obrotu osi enkodera (rysunek 26). Wyjście z funkcji następuje po naciśnięciu osi enkodera. Wybrany numer wejścia jest zapisywany w pamięci EEPROM i odtwarzany po włączeniu zasilania.

Funkcja „Ustawienia” zawiera podmenu umożliwiający ustawienie godziny, daty, określenie, czy godzina i data mają być wyświetlane na ekranie głównym oraz ustawienie jasności podświetlenia (rysunek 27).

Ustawianie czasu i daty jest wykonywane przez funkcje „Ustaw czas” i „Ustaw datę”. Obie działają podobnie. Po wejściu do nastaw godziny najpierw są ustawiane minuty przez kręcenie osią enkodera. Żądana wartość jest akceptowana przyciśnięciem osi i program automatycznie przechodzi do ustawiania

godzin przez kręcenie osią (rysunek 28). W momencie przyciśnięcia osi jest zerowany licznik sekund, do układu zegara DS1307 są transmitowane nastawy sekund, minut i godzin oraz jest uruchamiane zliczanie czasu. Ekran funkcji ustawiania daty pokazano na rysunku 29. Funkcja „Wysw. daty” włącza i wyłącza wyświetlanie czasu i daty na ekranie głównym w dolnej linijce wyświetlacza. Zostało to pokazane na rysunkach 30 i 31.

Ważną funkcją sterownika jest programowanie kodów „Kody RC5”. Po jej wywołaniu trzeba skierować nadajnik pilota w stronę odbiornika IR i nacisnąć przycisk odpowiadający wyświetlanej funkcji. Odebrany kod zostanie wyświetlony na ekranie zamiast czterech kresek (rysunek 32). Zatwierdzenie kodu i przejście do następnego jest wykonywane po naciśnięciu osi enkodera obrotowego. Funkcja kończy swoje działanie po zaprogramowaniu wszystkich kodów.

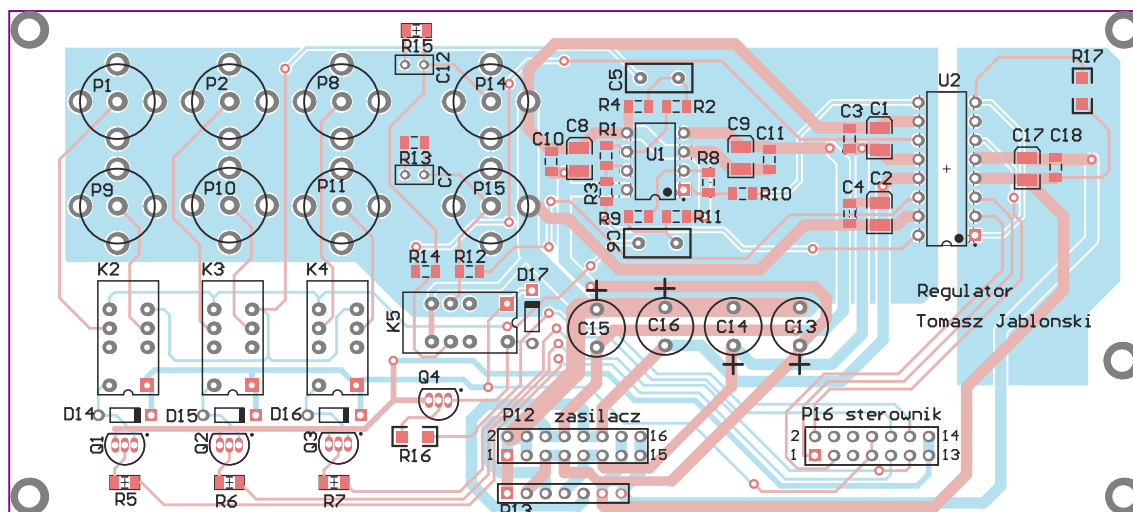
Listing 11. Wysłanie 16 bitów do PGA2311

```

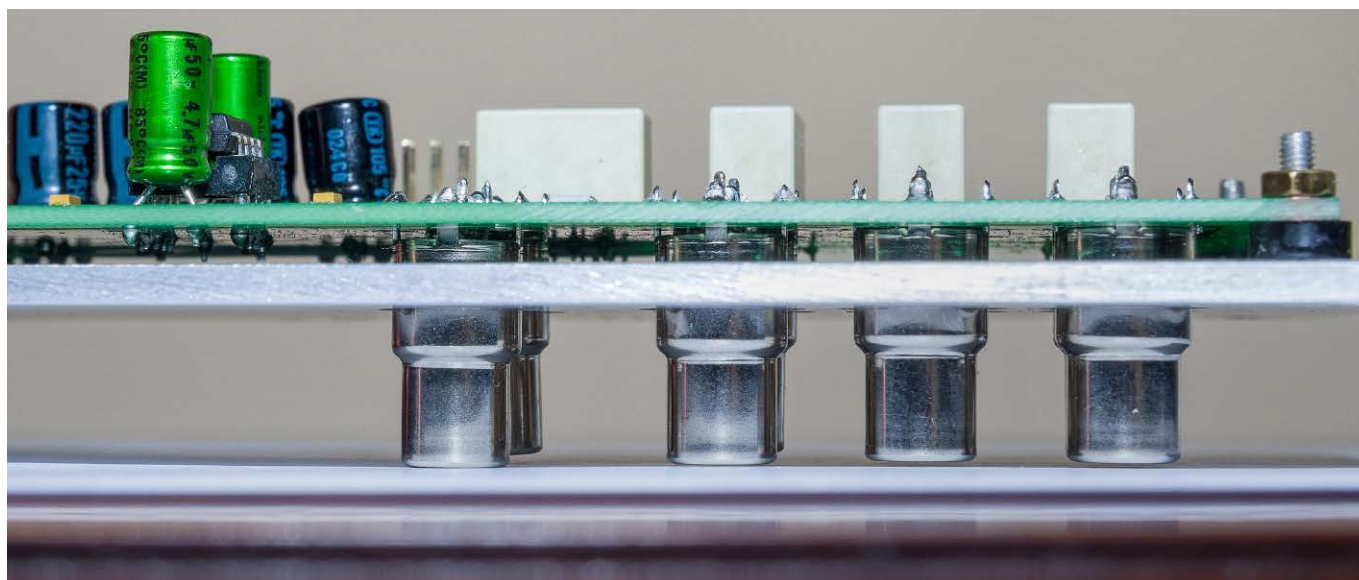
/*****
Wysłanie 16 bitów do PGA2311
*****/
void SendPGA(unsigned char volL, unsigned char volR)
{
    uint16_t vol=0;
    unsigned char i;

    vol=0;
    vol=volR;
    vol=(vol<<8)|volL;
    CS=0;
    del();
    for(i=0;i<16;i++)
    {
        SCK=0;
        del();
        if((vol&0x8000)==0) SDO=0;
        else SDO=1;
        vol=vol<<1;
        del();
        SCK=1;
        del();
    }
    SCK=0;
    del();
    SDO=0;
    del();
    CS=1;
    del();
}

```



Rysunek 20. Schemat montażowy płytki regulatora



Fotografia 21. Montaż złączy cinch

Listing 12. Funkcja menu głównego

```

/*****
Główne menu funkcyjne
*****/
void MenuFun(void){
    unsigned char fun;
    unsigned char kod;

    LcdClear(); //wyczyść wyświetlacz
    fun=DATAEE_ReadByte(ADR_FUN);
    PosLcd(4,1); PosLcd(4,1);
    DispLcd(„Zmien wejście”);
    PosLcd(4,2);
    DispLcd(„Ustawienia”);
    PosLcd(4,3);
    DispLcd(„Kody RC5”);
    PosLcd(4,4);
    DispLcd(„Wyświetlacz”);
    while(1){
        MarkFun(fun);
        kod=GetEncoder();
        if(kod==KOD_IMP_ST)
        {
            while(ST==0);
            DATAEE_WriteByte(ADR_FUN, fun); //ostatnio wybrana funkcja
            break;
        } //przyciśnięcie oski - koniec ustawiania
        if(kod==KOD_IMP_UP) //zwiększ wartość
        {
            ++fun;
            if(fun==4) fun=0; //ClearEnc();//wyczyść kody enkodera
        }
        if(kod==KOD_IMP_DWN) //zwiększ wartość
        {
            --fun;
            if(fun==0xff) fun=3;
        }
    }
    LcdClear(); //wyczyść wyświetlacz
    if(fun==0) UstawWejscie();
    if(fun==1) MenuFun();
    if(fun==2) SetCodeRC5();
    if(fun==3) UstawWysw();
    MainScreen();
}

```

Listing 13. Funkcja uczenia się kodów RC5

```

/*****
Uczenie kodów RC5
*****/
void SetCodeRC5(void)
{
    int kod;

    LcdClear();
    PosLcd(1,1);
    DispLcd(„kody RC5”);
    PosLcd(1,2);
    DispLcd(„DOWN ---- ”);
    while(IR==1); //czekaj na początek kodu
    while(1){
        kod=GetRc5();
        if(kod!=0xffff) break;
    }
    //RC5_DOWN=kod;
    DispHex((kod>>8),6,2); //kod rc5 w postaci hex
    DispHex(kod,8,2);
    DATAEE_WriteByte(ADR_RC5_DWN,kod>>8);
    DATAEE_WriteByte(ADR_RC5_DWN+1,kod);
    Delay_ms(200);
    while(ST==1); //czekaj na przyciśnięcie oski
    PosLcd(1,2);
    DispLcd(„ UP ---- ”);
    while(IR==1); //czekaj na początek kodu
    while(1){
        kod=GetRc5();
        if(kod!=0xffff)
            break;
    }
    //RC5_UP=kod;
    DispHex((kod>>8),6,2); //kod rc5 w postaci hex
    DispHex(kod,8,2);
    DATAEE_WriteByte(ADR_RC5_UP,kod>>8);
    DATAEE_WriteByte(ADR_RC5_UP+1,kod);
    while(ST==1); //czekaj na przyciśnięcie oski
    PosLcd(1,2);
}

```



Fotografia 22. Płytkę regulatora zamontowaną na panelu tylnym



Fotografia 23. Widok panelu tylnego



Rysunek 24. Ekran główny



Rysunek 25. Menu funkcyjne



Rysunek 26. Ekran funkcji wyboru aktywnego wejścia



Rysunek 27. Podmenu funkcji Ustawienia



Rysunek 28. Ustawianie czasu



Rysunek 29. Ustawianie daty



Rysunek 30. Włączenie wyświetlania czasu i daty na ekranie głównym



Rysunek 31. Ekran główny z wyświetlaniem czasu i daty



Rysunek 32. Programowanie kodów pilota RC5



Rysunek 33. Ustawianie jasności podświetlenia wyświetlacza

Ostatnią funkcją przewidzianą w tej wersji oprogramowania jest ustawienie jasności podświetlenia wyświetlacza LCD. Można to zrobić w zakresie od 0 do 10. Wartość 0 wyłącza podświetlenie, a 10 włącza je z pełną mocą. W praktyce dobre efekty zależnie od jasności pomieszczenia uzyskuje się w zakresie od 3 do 6. Wartości są ustawiane przez kręcenie osi. Jednocześnie zmienia się jasność podświetlenia. Po naciśnięciu osi ustawiona wartość jest zapisywana w pamięci EEPROM, a funkcja kończy swoje działanie (rysunek 33).

Podsumowanie

Opisywany tu przedwzmacniacz od jakiegoś czasu jest użytkowany bez najmniejszych problemów eksploatacyjnych. Głównym źródłem sygnału jest opisywany przeze mnie na łamach „Elektroniki Praktycznej” przetwornik cyfrowo-analogowy z układami Wolfsona WM8741 oraz stary tuner FM

Listing 13. cd.

```

DispLcd(„FUN ---- „);
while (IR==1); //czekaj na poczatek kodu
while (1) {
    kod=GetRc5();
    if (kod!=0xffff)
        break;
}
//RC5_FUN=kod;
DispHex((kod>>8),6,2); //kod rc5 w postaci hex
DispHex(kod,8,2);
DATAEE_WriteByte(ADR_RC5_FUN,kod>>8);
DATAEE_WriteByte(ADR_RC5_FUN+1,kod);
while (ST==1); //czekaj na przycisniecie oski
PosLcd(1,2);
DispLcd(„MUT ---- „);
while (IR==1); //czekaj na poczatek kodu
while (1) {
    kod=GetRc5();
    if (kod!=0xffff) break;
}
//RC5_MUT=kod;
DispHex((kod>>8),6,2); //kod rc5 w postaci hex
DispHex(kod,8,2);
DATAEE_WriteByte(ADR_RC5_MUT,kod>>8);
DATAEE_WriteByte(ADR_RC5_MUT+1,kod);
while (ST==1); //czekaj na przycisniecie oski
PosLcd(1,2);
DispLcd(„IN1 ---- „);
while (IR==1); //czekaj na poczatek kodu
while (1) {
    kod=GetRc5();
    if (kod!=0xffff) break;
}
//RC5_MUT=kod;
DispHex((kod>>8),6,2); //kod rc5 w postaci hex
DispHex(kod,8,2);
DATAEE_WriteByte(ADR_RC5_IN1,kod>>8);
DATAEE_WriteByte(ADR_RC5_IN1+1,kod);
while (ST==1); //czekaj na przycisniecie oski
PosLcd(1,2);
DispLcd(„IN2 ---- „);
while (IR==1); //czekaj na poczatek kodu
while (1) {
    kod=GetRc5();
    if (kod!=0xffff) break;
}
//RC5_MUT=kod;
DispHex((kod>>8),6,2); //kod rc5 w postaci hex
DispHex(kod,8,2);
DATAEE_WriteByte(ADR_RC5_IN2,kod>>8);
DATAEE_WriteByte(ADR_RC5_IN2+1,kod);
while (ST==1); //czekaj na przycisniecie oski
Delay_ms(20);
while (ST==0); //czekaj na puszczenie oski
DATAEE_WriteByte(ADR_RC5_ZN,1); //znacznik zaprogramowania kodow
GetRc5Code();
}
    
```

firmy Sansui. Od czasu do czasu, do jednego z wejść jest przyłączane wyjście słuchawkowe dość leciwego smartfona i są odtwarzane pliki MP3. Wyjście przedwzmacniacza steruje dwoma monoblokami wzmacniacza Ugoda. Ugoda jest znaną fanom DIY audio bardzo dobrą konstrukcją wzmacniacza na tranzystorach bipolarnych.

W czasie eksploatacji okazało się, że podstawowe funkcje przedwzmacniacza działają poprawnie. Nie ma problemów z zawieszaniem

się wyświetlacza lub całego sterownika. W przyszłości jest planowane dopisanie funkcji regulowania balansu, a przede wszystkim funkcji programowanego offsetu wzmocnienia przypisanego do wejść 2 i 3. Offset będzie programowany w odniesieniu do bazowego poziomu wejścia 1. Pozwoli to na wyrównanie poziomów sygnałów pomiędzy wyjściem przetwornika i wyjściem tunera FM.

Tomasz Jabłoński, EP