

Zasilacz warsztatowy

AVT 5417

Regulowany zasilacz warsztatowy to – oprócz lutownicy, multimetru, oscyloskopu i generatora – podstawowe wyposażenie warsztatu elektronika. Nawet początkujący elektronik musi być wyposażony przynajmniej w zasilacz, lutownicę i multimetr. Dlatego wielu z nich konstruuje swoje pierwsze zasilacze samodzielnie, zanim „dojrzeje” do zakupu wyrobu fabrycznego. Postanowiłem i ja zmierzyć się z konstrukcją zasilacza, jednak używając do tego celu nowoczesnych podzespołów. Wyposażyłem go też w funkcje, które uważałem za niezbędne.

Rekomendacje: pełnowartościowy, nowoczesny, funkcjonalny zasilacz o regulowanym napięciu, prądzie wyjściowym i prądzie zadziałania zabezpieczenia, który przyda się w niejednym warsztacie elektronicznym.

Od czasu, gdy w handlu pojawiły się względnie tanie mikrokontrolery, konstruktorzy próbowali je z lepszym lub gorszym skutkiem użyć jako sterowników zasilaczy. Niedługo wykonałem dla siebie zasilacz sterowany przez mikrokontroler PIC16F84. Była to konstrukcja, w której rolę stabilizatora pełnił układ L200. Regulacja napięcia wyjściowego polegała na dołączaniu poprzez analogowy multiplexer w obwód regulacji L200 potencjometrów z ustalonym podzia-

łem, a w konsekwencji ustalenie napięcia wyjściowego. Multiplexer miał 16 wyjść, więc do dyspozycji miałem 16 napięć wyjściowych. Dodatkowym ograniczeniem było dopuszczalne napięcie na wyprowadzeniach multiplexera wynoszące +16 V. Dlatego napięcie wyjściowe nie mogło być wyższe niż ok. 15 V.

Ograniczenie prądowe w L200 jest realizowane przez włączenie w obwód pomiaru prądu rezystora szeregowego. W moim zasi-

W ofercie AVT*

AVT-5417 A

Podstawowe informacje:

- Zakres napięcia wyjściowego 0,6...24 V.
- Napięcie wyjściowe ustawiane zgrubnie, z krokiem 1 V i dokładnie z krokiem 0,1 V.
- Funkcja programowego ograniczania maksymalnego napięcia wyjściowego – overvoltage protection.
- Maksymalny prąd wyjściowy 3 A.
- Programowane ograniczenie prądowe w zakresie 0,1...3 A.
- Ograniczenie prądowe ustawiane zgrubnie z krokiem 1 A i dokładnie z krokiem 0,1 A.
- Funkcja programowanego zabezpieczenia nadprądowego (overcurrent protection) w zakresie 0,1...3 A ustawiana z rozdzielczością 0,1 A.
- Cyfrowy pomiar napięcia wyjściowego (rozdzielczość 0,1 V) i prądu wyjściowego (rozdzielczość 0,01 A).
- Funkcja ograniczania moc strat na tranzystorze regulacyjnym przez automatyczne przełączanie odczepu transformatora na połowę wtórnego napięcia przemiennego przy napięciu wyjściowym poniżej +12 V (z histerezą 0,2 V).
- Możliwość pracy bez sterownika mikroprocesorowego – sterowanie za pomocą potencjometrów (bez pomiaru napięcia i prądu wyjściowego).

Dodatkowe materiały na CD lub FTP:

<http://ep.com.pl>, user: 41650, pass: 742qofb6

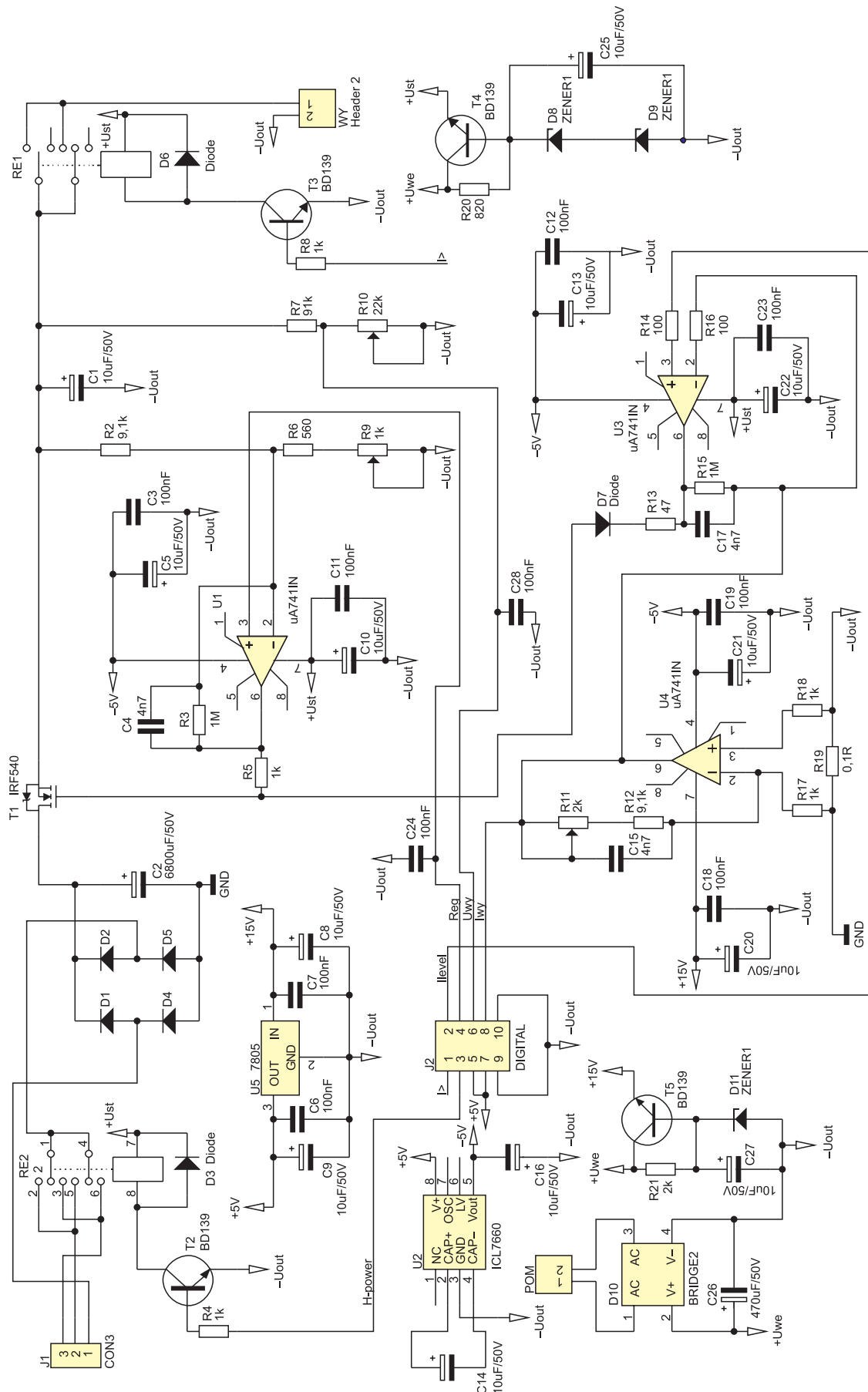
- wzory płytek PCB
- karty katalogowe i noty aplikacyjne elementów oznaczonych w Wykazie elementów kolorem czerwonym

* Uwaga:
Zestawy AVT mogą występować w następujących wersjach:
AVT xxxx UK to zaprogramowany układ. Iylko i wyłącznie. Bez elementów dodatkowych.
AVT xxxx A płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.
AVT xxxx A+ płytka drukowana i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.
AVT xxxx B płytka drukowana (lub płytki) oraz komplet elementów wymienionych w załączniku pdf
AVT xxxx C to nic innego jak zmontowany zestaw B, czyli elementy wlotowane 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>

laczu, za pomocą przełączników, włączałem 3 rezystory uzyskując 3 zakresy ograniczenia prądowego: 100 mA, 500 mA i 1000 mA.

Napięcie wyjściowe było ustawiane za pomocą przycisków „Góra” i „Dół”. Sterownik wyświetlał na wyświetlaczu jego wartość,

przy czym nie była to wartość zmierzona na wyjściu, tylko spodziewana. W procesie uruchamiania trzeba było zadbać o to, aby



Rysunek 1. Schemat części analogowej zasilacza

wartość wyświetlana odpowiadała wyjściowej przez odpowiednie ustawienie potencjometru włączanego w obwód. Zabezpieczenie prądowe było zmieniane sekwencyjnie przez jednoczesne przyciśnięcie obu przycisków. Piszę o tym wszystkim dlatego, że ten zasilacz (opisywany kiedyś na łamach EP) pracuje bezawaryjnie kilkanaście lat i jest do dzisiaj z powodzeniem używany mimo wspomnianych ograniczeń. Teraz postanowiłem skonstruować i wykonać nowszą wersję, o szerszym zakresie napięcia wyjściowego, z funkcją pomiaru prądu i napięcia wyjściowego oraz z regulowanym zabezpieczeniem prądowym.

Założenia projektowe

Zasilacze mogą mieć bardzo różne parametry, ale najczęściej przyjmuje się, że uniwersalny zasilacz warsztatowy powinien mieć możliwość regulacji napięcia wyjściowego w zakresie 0...24-30 V i wydajność prądową 2...5 A. Dolna granica 0 V jest umowna, ponieważ komu nie jest potrzebny zasilacz, który nie daje napięcia na wyjściu. W praktyce używa się napięcia minimalnego rzędu 1 V.

Ja przyjąłem, że zadowolający będzie zakres napięcia wyjściowego 0,6...24 V i wydajność prądowa 3 A. Założyłem też, że urządzenie będzie kontrolowane przez sterownik mikroprocesorowy, który będzie mierzył rzeczywiste napięcie wyjściowe oraz prąd obciążenia. Pomiar prądu umożliwi wykonanie zabezpieczenia nadprądowego polegającego na odłączeniu napięcia zasilającego od obciążenia po przekroczeniu zadanego progu. Dodatkowo, dobrze by było zastosować regulowane ograniczenie prądowe. Po osiągnięciu zadanej wartości prądu jest ograniczane napięcie wyjściowe. Zasilacz pracuje wtedy jako źródło prądowe.

Układ ma być klasycznym stabilizatorem analogowym z szeregowym tranzystorem regulacyjnym. Przy takiej topologii układu występują duże straty mocy na szeregowym stabilizatorze przy obciążeniu dużym prądem i ustalonym, niewielkim napięciu wyjściowym. Dlatego przewidziałem automatyczne przełączenie zasilania układu z odczepu transformatora sieciowego, gdy ustalone napięcie wyjściowe jest niższe niż 12 V. Sterowanie napięciem wyjściowym, ograniczeniem prądowym, zabezpieczeniem nadprądowym oraz pomiar napięcia i prądu ma wykonywać sterownik z mikrokontrolerem.

Część analogowa zasilacza

Schemat części analogowej zasilacza pokazano na **rysunku 1**. Na pierwszy rzut oka układ wydaje się dość skomplikowany. Jednak w rzeczywistości jest to kilka klasycznych bloków analogowych połączonych ze sobą:

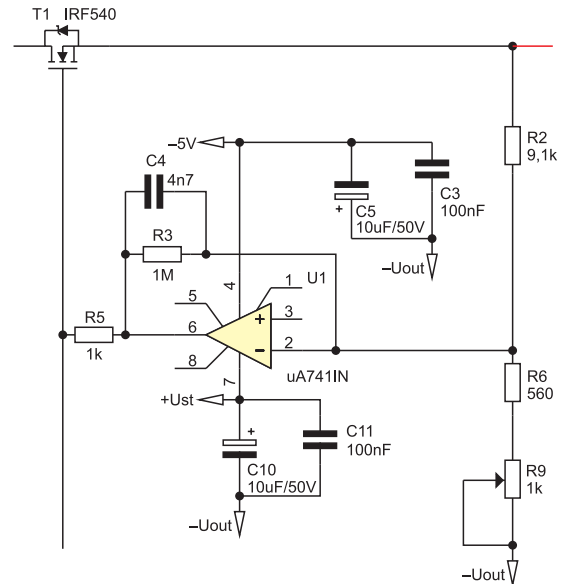
- Blok regulatora napięcia.
- Blok pomiaru prądu i napięcia.
- Blok ograniczenia prądowego.
- Blok zasilacza pomocniczego.

Blok regulatora napięcia.

Blok regulatora napięcia (**rysunek 2**) jest zbudowany z tranzystora regulacyjnego T1, komparatora i wzmacniacza błędów U1 oraz układu próbkującego napięcie wyjściowe. Tranzystorem regulacyjnym T1 może być tranzystor bipolarny NPN lub tranzystor FET z kanałem N. Zasada działania tego układu jest następująca: napięcie wyjściowe ze źródła (emitera) tranzystora T1 po podzieleniu przez 10 w układzie dzielnika złożonego z rezystorów R2, R6 i R9 (sygnał regulacji) trafia na wejście odwracające wzmacniacza operacyjnego U1 (nóżka 2). Na wejście nieodwracające U1 (nóżka 3) jest podawane napięcie odniesienia nazywane napięciem referencyjnym Uref. W wyniku porównania sygnału regulacji z Uref powstaje sygnał błędów. Sygnał błędów jest wzmacniany i z wyjścia U1 (nóżka 6) steruje bramką (bazą) tranzystora regulacyjnego T1. Napięcie wyjściowe jest ciągle testowane i jeżeli nie jest równe zadanemu, czyli napięciu referencyjnemu, to sygnał z wyjścia komparatora po wzmocnieniu tak steruje tranzystorem T1, by te różnicę zniwelować.

Układ regulacji napięcia można traktować jako wzmacniacz operacyjny mocy objęty pętlą ujemnego sprzężenia zwrotnego poprzez dzielnik napięcia wyjściowego (rezystory R1, R6, R9). Napięcie na wyjściu można wyliczyć z zależności $U_{wy} = (1 + R2 / (R6 + R9)) \cdot U_{ref}$. Ponieważ założyliśmy, że napięcie wyjściowe jest dzielone przez 10, czyli $R2 / (R6 + R9) = 10$, to napięcie wyjściowe ma wartość $U_{wy} = 10 \cdot U_{ref}$. Wystarczy teraz zmieniać napięcie referencyjne w zakresie 0...2,4 V, by na napięcie wyjściowe zmieniało się w zakresie 0...24 V. Zmiana napięcia referencyjnego będzie zadaniem sterownika mikroprocesorowego.

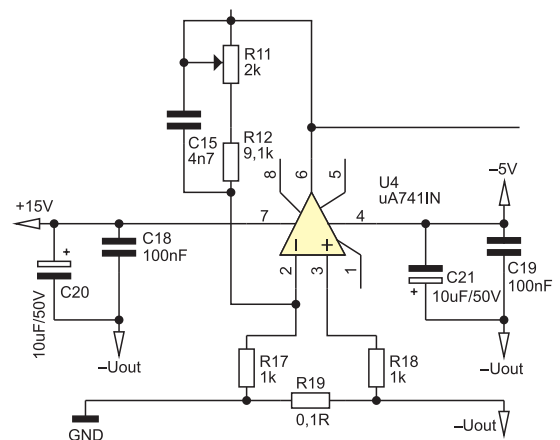
Układ regulacji z ujemnym sprzężeniem zwrotnym ma bardzo dobre parametry elektryczne i dlatego jest chętnie stosowany w topologiach scalonych stabilizatorów napięcia oraz w regulatorach wykonywanych z elementów dyskretnych. Żeby ograniczyć możliwość wzbudzenia się układu ograniczono pasmo przenoszenia wzmacniacza U1 przez dodanie rezystora R3 i kondensatora C4.



Rysunek 2. Blok regulatora napięcia

Blok pomiaru napięcia i prądu. Napięcie wyjściowe regulatora napięcia będzie mierzone przez wbudowany w mikrokontroler przetwornik A/C. Ponieważ mikrokontroler jest zasilany napięciem +3,3 V i to napięcie jest używane jako napięcie referencyjne przetwornika, to sygnał na wejściu przetwornika nie może mieć napięcia wyższego niż +3,3 V. Najprościej będzie podzielić napięcie wyjściowe przez 10. Dzielnik jest zbudowany z rezystora R7 i potencjometru R10. Zastosowałem wielobrotowy, precyzyjny potencjometr, bo jest to łatwiejsze rozwiązanie niż zdobycie i dobranie rezystorów do dokładnego dzielnika.

Pomiar prądu (**rysunek 3**) nie jest tak prosty, jak pomiar napięcia. Najczęściej natężenie prądu mierzy się metodą techniczną poprzez pomiar spadku napięcia na rezystorze pomiarowym włączonym szeregowo. Powinien on mieć jak najmniejszą rezystancję po to, aby nie powodował dużego spadku napięcia wyjściowego przy dużym prądzie obciążenia zasilacza. Z drugiej strony, małe spadki napięcia na rezystorze pomiarowym powodują problemy z pomiarem. Dlatego stosuje się rezystory pomiarowe o małej re-



Rysunek 3. Układ pomiaru prądu

zstancji i wzmacnianie spadku napięcia poprzez wzmacniacze prądu stałego. Prąd płynący przez rezystor wywołuje na nim spadek napięcia, który z prawa Ohma jest równy $U=I \times R$ ($R=R19$). Przy znanej rezystancji R prąd będzie równy $I=U/R$. Załóżmy, że nasz zasilacz będzie dostarczał prądu o natężeniu maksymalnym 3 A, a boczniak będzie miał rezystancję 0,1 Ω . Maksymalny spadek napięcia $U=3 \text{ A} \times 0,1 \Omega=0,3 \text{ V}$. Z jednej strony, 300 mV spadku na wyjściu to sporo jak na zasilacz, ale z drugiej strony nie będzie można dokładnie zmierzyć to napięcie przetwornikiem o zakresie napięcia wejściowego 0...3,3 V (napięcie referencyjne przetwornika A/C wynosi 3,3 V). Napięcie 300 mV to około 10% całego zakresu pomiarowego, a trzeba dążyć do tego, by napięcie mierzone miało maksymalną wartość bliską zakresowi pomiarowemu.

Najprostszym rozwiązaniem będzie wzmocnienie napięcia stałego odkładające się na boczniku za pomocą wzmacniacza operacyjnego U4 pracującego w konfiguracji wzmacniacza nieodwracającego. Wzmocnienie takiego układu łatwo dobiera się za pomocą dzielnika złożonego z 2 rezystorów: R17 i sumy rezystancji R11+R12.

Wzmocnienie wzmacniacza nieodwracającego jest wyliczane z zależności: $G=1+(R11+R12)/R17$. Jeżeli dobierzemy rezystory w taki sposób, aby $R11+R12/R17$ miało wartość 9, to wzmocnienie całego układu będzie równe 10. Wtedy dla maksymalnego natężenia prądu wynoszącego 3 A napięcie wyjściowe z układu pomiaru prądu będzie równe 3 V. Dokładne ustalenie wzmocnienia jest możliwe dzięki zastosowaniu wieloobrotowego potencjometru R11. Pozwala to również na korekcję wzmocnienia wynikającą z tolerancji rezystancji użytego rezystora pomiarowego.

Rezystor pomiarowy jest włączony w ujemną gałąź napięcia wyjściowego. Uproszcza to układ pomiarowy, ale wymaga uwagi przy projektowaniu obwodów masy. Masa odniesienia w stosunku, do której są mierzone napięcia wyjściowe i zasilane są wzmacniacze operacyjne, jest oznaczona jako -Uout. Jak widać na schemacie, nie jest to ujemny biegun mostka Graetza, tylko drugi koniec rezystora pomiarowego R19. Takie rozwiązanie jest podyktowane tym, aby układ próbkowania bloku regulacji napięcia pomiarowego nie mierzył dodatkowego spadku na rezystorze R19. Gdyby masa układu próbkowania była dołączana do minusa mostka (oznaczonego na schemacie GND), to napięcie wyjściowe regulatora zmieniłoby się o wartość spadku napięcia na R19 i było zależne od pobieranego prądu (w granicach 0...300 mV dla prądu wyjściowego 0...3 A), co jest niedopuszczalne. -Uout jest też masą odniesienia dla sygnałów regulacyjnych generowanych przez mikrokontroler.

Blok ograniczenia prądowego. Ograniczenie prądowe to istotny obwód zasilacza. Prawidłowo ustawione ograniczenie często chroni zasilany układ przed zniszczeniem, szczególnie wtedy, gdy wydajność prądowa zasilacza jest spora, a zasilane urządzenie ma np. odwrotnie wlotowany układ scalony, kondensator elektrolityczny itp. Ograniczenie prądowe chroni też sam zasilacz przed krótkotrwałym zwarciem wyjścia.

Zasada działania układu ograniczenia prądowego jest podobna do działania układu regulacji napięcia. Jego podstawą jest komparator ze wzmacniaczem napięcia. Komparator porównuje napięcie wyjściowe z układu wyżej opisanego układu pomiaru prądu z napięciem referencyjnym reprezentującym ograniczany prąd. Zastosowany w zasilaczu układ ograniczenia został pokazano na **rysunku 4**.

Na wejście nieodwracające (nóżka 3) przez rezystor R14 jest podawane ze sterownika mikroprocesorowego napięcie referencyjne. Wejście odwracające jest połączone przez rezystor R16 z wyjściem układu pomiaru prądu z rysunku 3 (nóżka 6 układu U4). Jeżeli napięcie na wejściu odwracającym reprezentujące mierzony prąd na wyjściu zasilacza jest wyższe niż napięcie referencyjne podawanego z mikrokontrolera na wejście nieodwracające, to na wyjściu wzmacniacza U3 występuje napięcie niższe, niż na wyjściu układu regulacji napięcia. Powoduje to zmniejszanie się napięcia na wyjściu zasilacza do momentu aż napięcia na wejściu komparatora się wyrównają. Dioda D7 zabezpiecza układ regulacji przez podwyższaniem napięcia wyjściowego przez układ zabezpieczenia, kiedy prąd wyjściowy jest niższy niż prąd ograniczenia.

Blok zasilania pomocniczego. Układy wzmacniaczy operacyjnych zastosowane w zasilaczu muszą być zasilane odpowiednim napięciem. Wzmacniacze operacyjne U1 i U3 są jednocześnie wzmacniaczami napięcia dla układu regulacji napięcia i ograniczenia prądowego. Z tego powodu muszą być zasilane napięciem o wartości co najmniej $U_{wy}+3 \text{ V}$, czyli +27 V. Dlatego zastosowano wysokonapięciową wersję typowego wzmacniacza $\mu\text{A}741$ oznaczoną jako $\mu\text{A}741\text{IN}$. Układ ten może pracować przy maksymalnym napięciu zasilania $\pm 22 \text{ V}$, a więc 0...44 V. W zasilaczu jest zasilany napięciem -5 V...+27 V (32 V). $\mu\text{A}741$ nie może pracować z ujemnym biegunem zasilania na potencjale masy (tak jak na przykład LM324) i napięciami wejściowymi równymi lub bliskimi potencjałowi masy. Do generowania napięcia ujemnego -5 V jest używany popularny układ ICL7660 (U2).

Napięcie +27 V jest uzyskiwane w stabilizatorze zbudowanym z tranzystora T4 i diod Zenera D8 (15 V), oraz

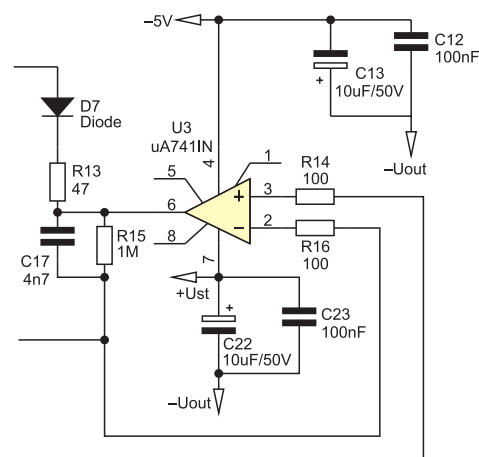
D9 (12 V). Wzmacniacz U4 jest zasilany napięciem +15 V ze stabilizatora zbudowanego z tranzystora T5 i diody Zenera D11. Tranzystor T5 powinien być wyposażony w radiator. Stabilizator 7805 (U5) dostarcza napięcia +5 V do zasilania układów sterownika, w tym mikrokontrolera (przez stabilizator +3,3 V umieszczony na płytce sterownika) i wyświetlacza LCD. Układy zasilania pomocniczego wymagają napięcia niestabilizowanego o wartości ok. +32 V. Do złącza POM trzeba doprowadzić dodatkowe napięcie przemiennie o wartości ok. 32...35 V o wydajności minimum 150 mA.

Układ stabilizatorów zasilania pomocniczego można próbować zasilić z napięcia na dodatnim biegunie kondensatora C2. Jednak to rozwiązanie ma ograniczenia. Po pierwsze, napięcie na C2 musi być odpowiednio wyższe, niż maksymalne napięcie wyjściowe zasilacza. Ten warunek można dość łatwo spełnić, ale wraz ze wzrostem napięcia na C2, rosną straty na tranzystorze regulacyjnym. Ponadto, przy takim rozwiązaniu nie można wykorzystać metody włączania odciążenia transformatora w celu zmniejszania strat mocy na tranzystorze regulacyjnym, gdy jest ustawione napięcie wyjściowe równe połowie maksymalnego napięcia zasilacza. Wtedy napięcie na C2 jest zbyt niskie. Jednak podstawowe ograniczenie to odkładanie się tętnień o częstotliwości 100 Hz na rezystorze pomiarowym R19. Po 10-krotnym wzmocnieniu, na wyjściu układu pomiaru prądu tętnienia mogą mieć amplitudę nawet powyżej 2 V.

Sterownik cyfrowy

Według założeń sterownik cyfrowy powinien wykonywać następujące zadania:

- Wytwarzać napięcie referencyjne w zakresie 0...2,4 V do regulacji napięcia wyjściowego.
- Wytwarzać napięcie regulujące poziom ograniczenia prądowego w zakresie 0...3 V.
- Mierzyć napięcie wyjściowe.



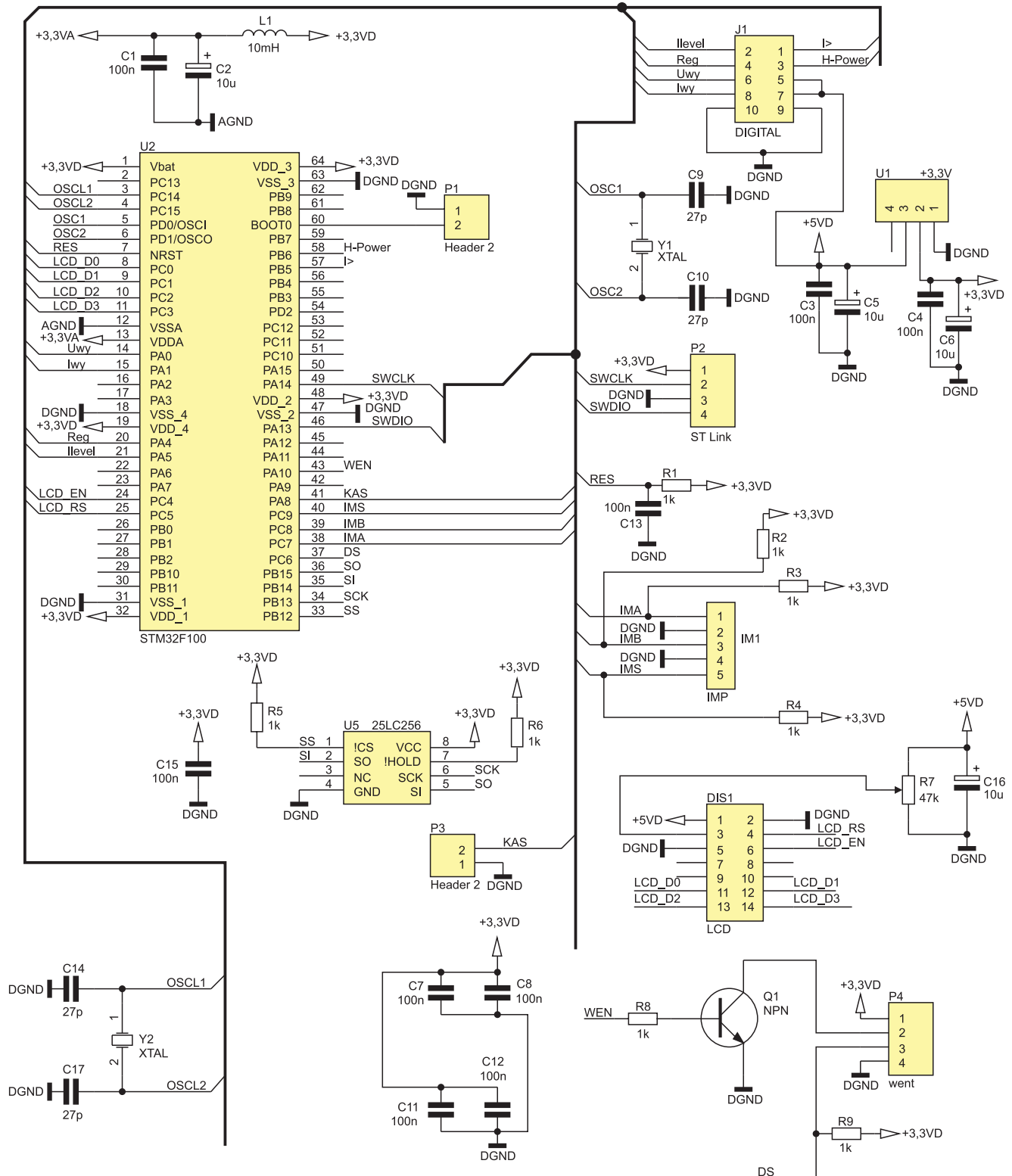
Rysunek 4. Układ ograniczenia prądowego

- Mierzyć prąd wyjściowy z konwertera prąd/napięcie.
- Realizować interfejs użytkownika umożliwiający wykonywanie wszystkich nastaw i pomiarów.

Do wytwarzania napięcia przez cyfrowy sterownik będzie potrzebny przetwornik C/A. Takie przetworniki jako moduły peryferyjne do niedawna były rzadko spotykane w mikrokontrolerach. Jeżeli trzeba było wytworzyć napięcie, to najprostszym rozwiązaniem

był przetwornik C/A zbudowany z generatora PWM i analogowego filtra dolnoprzepustowego. Przebieg PWM można bez problemu generować nawet programowo, ale filtr musi być układem zewnętrznym, co komplikuje układ. Z tego powodu zacząłem szukać mikrokontrolerów z wbudowanym przetwornikiem C/A. Poszukiwania były o tyle trudne, że tu potrzebuję dwóch niezależnych przetworników. Wybór padł na 32-bitowy mikrokontroler STM32F100RBT6B. Jest on

bogato wyposażony w bloki peryferyjne, ma obszerne pamięci Flash i RAM, a wydajność 32-bitowego rdzenia Cortex-M zdecydowanie przewyższa potrzeby tej aplikacji. Ale ma również wszystkie potrzebne do budowy zasilacza zasoby: dwa 12-bitowe przetworniki C/A i 12-bitowy przetwornik A/C z multiplexerem na wejściu i co ważne – jest łatwo dostępny, a jego cena przy zakupie jednostkowym jest nie jest wygórowana. Zatem nie stoi na przeszkodzie, by go zastosować,



Rysunek 5. Schemat ideowy sterownika zasilacza

choć z zadaniem sterowania zasilacza poradziłby sobie nawet niezbyt rozbudowany 8-bitowiec wyposażony w odpowiednie bloki peryferyjne. Schemat sterownika pokazano na **rysunku 5**.

Zasadniczy element – mikrokontroler – jest zasilany napięciem +3,3 V uzyskiwanym ze stabilizatora U1. Na wejście stabilizatora jest podawane napięcie +5 V ze stabilizatora U5 (7805) umieszczonego na płycie analogowej. To napięcie zasilają również wyświetlacz LCD (DIS1). Taktowanie zapewnia rezonator kwarcowy o częstotliwości 8 MHz (Y2).

Układy analogowe (zasilanie +3,3 V) nie mają swojego stabilizatora. Żeby zapewnić przyzwoitą jakość napięcia zasilającego układu analogowego, w tym układy wytwarzania napięć referencyjnych przetworników A/C i C/A, zastosowano filtr LC składający się z dławika L1 oraz kondensatorów C1 i C2. Ponieważ mikrokontroler nie ma wbudowanej pamięci nieulotnej przeznaczonej na dane użytkownika (na przykład dane konfiguracyjne), to zastosowałem zewnętrzną pamięć EEPROM typu 25LC256 (U5) z interfejsem SPI.

Interfejs użytkownika składa się z alfanumerycznego wyświetlacza LCD o organizacji 2×16 znaków (DIS1), impulsatora ze stykiem zwierzanym przyciśnięciem ośki (IM!) i dodatkowego styku KAS (P3). Programowanie mikrokontrolera zapewnia złącze SWI (P2) obsługiwane przez programator/debuger ST-LINK.

Elementy R8, Q1 i P4 nie są używane w projekcie. Zostały przewidziane do sterowania wentylatorem chłodzącym radiator i odczytywania temperatury radiatora.

Jak widać sterownik układowo nie jest skomplikowany. Uzyskanie tak nieskomplikowanego układu było możliwe dzięki bogatemu wyposażeniu mikrokontrolera.

Wytwarzanie napięcia referencyjnego – konfigurowanie przetwornika C/A

Napięcia referencyjne do regulacji napięcia wyjściowego i zabezpieczenia nadprądowego są wytwarzane przez blok przetwornika C/A. W STM32F100 może on być skonfigurowany do pracy z rozdzielczością 12 lub 8 bitów i ma 2 niezależne konwertery, a w praktyce są to 2 niezależne przetworniki. Ich wyjścia są dostępne na wyprowadzeniach PA4 (DAC1) i PA5 (DAC2). Napięcie na PA4 jest używane do regulowania napięcia wyjściowego zasilacza, a na PA5 do ustawiania poziomu ograniczenia prądowego. Zależnie od

wybranej rozdzielczości, cyfrowe dane do konwersji są wpisywane do:

- 8 najmłodszych bitów rejestru DAC_DHR8Rx[7:0].
- 12 bitów dosuniętych do lewej w rejestrze DAC_DHR12Lx[15:4].
- 12 bitów dosuniętych do prawej w rejestrze DAC_DHR12Rx[11:0].

Dane z rejestrów DAC_DHR są automatycznie zapisywane do rejestrów przetwornika DAC_DOR w czasie jednego cyklu magistrali APB1, ale tylko w wypadku, gdy użytkownik nie wybrał sprzętowego wyzwalania konwersji. Kiedy takie wyzwalanie zostało wybrane i warunek wyzwalania nastąpił, to przepisanie pomiędzy DAC_DHR i DAC_DOR zajmuje 3 cykle magistrali APB1.

Po zapisaniu DAC_DOR napięcie na wyjściu zmienia się po czasie Tsettleing zależnym od napięcia zasilania i obciążenia wyjścia analogowego przetwornika. Warto tu wspomnieć, że do wyjścia można dołączyć wbudowany w układ mikrokontrolera bufor analogowy i uzyskując stosunkowo niską impedancję wyjściową. W zasilaczu programowo wybrano buforowanie wyjścia.

Napięcie wyjściowe przetwornika pracującego w rozdzielczości 12-bitowej jest określone zależnością: $DAC_{out} = U_{ref} (DOR/4095)[V]$, gdzie U_{ref} – napięcie referencyjne przetwornika, DOR – zawartość rejestru DAC_DOR. Dla napięcia referencyjnego 3,3 V napięcie na wyjściu zmienia się w zakresie 0...3,3 V.

Do konfiguracji przetwornika zostaną użyte funkcje standardowej biblioteki CMSIS dla mikrokontrolerów STM32. Jak wiemy, do sterowania napięciem wyjściowym zasilacza zostanie wykorzystany przetwornik DAC1 z wyjściem napięciowym na wyprowadzeniu PA4 (Reg), a do ustawiania poziomu zabezpieczenia prądowego DAC2 z wyjściem napięciowym na wyprowadzeniu PA5 (Ilevel). W pierwszym kroku zdefiniujemy PA4 i PA5 jako wyprowadzenia analogowe (**listing 1**).

Aby przetwornik pracował tak jak tego od niego oczekujemy musi być odpowiednio skonfigurowany. Najpierw wyłączamy możliwość sprzętowego wyzwalania. Napięcie na wyjściu ma się zmienić po zapisaniu nowej wartości do rejestru przetwornika przez pro-

gram użytkownika. Potem trzeba zapewnić taktowanie modułom ADC1 i ADC2 z magistrali APB1 oraz trzeba je włączyć przez wywołanie funkcji `DAC_Cmd(DAC_Channel_1, ENABLE);` i `DAC_Cmd(DAC_Channel_2, ENABLE);`. Procedurę konfiguracji pokazano na **listingu 2**.

Dla modułu DAC1 zapis rejestru DAC_DHR i w rezultacie DAC_DOR wykonuje funkcja `DAC_SetChannel1Data(DAC_Align_12b_R, 2048);`. Jej argumentami są:

- Wielkość i format danych do konwersji. My będziemy wpisywali dane 12-bitowe dosunięte do prawej.
- Cyfrowa wartość DOR do konwersji na napięcie analogowe według zależności $U_{wy} = U_{ref} (DOR/4095)[V]$.

Tak samo są zapisywane rejestry modułu DAC2.

Z praktycznego punktu widzenia mamy wszystko, co jest konieczne do konwersji cyfrowo-analogowej i tym samym programowego sterowania wytwarzaniem napięcia na wyjściu konwertera C/A w zakresie 0...3,3 V.

Pomiar napięcia i prądu wyjściowego – konfigurowanie przetwornika A/C

W mikrokontrolerach rodziny STM32F100 są wbudowane dwa niezależne przetworniki A/C o rozdzielczości 12 bitów. Napięcie podawane na wejście analogowe można przypisać do jednej z dwóch grup: podstawowej (*regular channel*) i grupy nazwanej *injected channel*. Grupa *injected* ma wysoki priorytet wyzwalania konwersji. Jeżeli pojawi się sygnał wyzwalania dla tej grupy kanałów, to zostanie zatrzymane przetwarzanie w grupie *regular* (po zakończeniu cyklu) i przetwornik będzie przetwarzał sygnały z wejść grupy *injected*. Ten mechanizm został stworzony po to, by zapewnić jak najszybszą konwersję dla sygnałów krytycznych z punktu widzenia algorytmów przetwarzania sygnałów.

Oprócz priorytetu są jeszcze inne różnice. Grupa *injected* może przetwarzać sygnały z maksymalnie 4 wejść, ale do każdego kanału jest przypisany osobny rejestr wyniku. Do grupy *regular* można przypisać maksymalnie 16 wejść, a rejestr wyniku jest jeden wspólny.

Listing 1. Konfiguracja wyprowadzeń PA4 i PA5

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE); //włączenie taktowania GPIOA
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4 | GPIO_Pin_5 ;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
GPIO_Init(GPIOA, &GPIO_InitStructure);
```

Listing 2. Konfiguracja przetwornika

```
void DACInit(void)
{
    DAC_InitTypeDef DAC_InitStructure;
    DAC_InitStructure.DAC_Trigger = DAC_Trigger_None; //bez wyzwalania sprzętowego
    DAC_InitStructure.DAC_OutputBuffer = DAC_OutputBuffer_Enable; //dołączenie bufora analogowego
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_DAC, ENABLE); //włączenie taktowania
    DAC_Init(DAC_Channel_1, &DAC_InitStructure);
    DAC_Cmd(DAC_Channel_1, ENABLE); //włączenie DAC1
    DAC_Init(DAC_Channel_2, &DAC_InitStructure);
    DAC_Cmd(DAC_Channel_2, ENABLE); //włączenie DAC2
}
```

W naszym zasilaczu, przy stosunkowo dużej wydajności rdzenia i szybkich przetwornikach, spokojnie możemy użyć grupy *regular* do pomiaru napięcia i prądu wyjściowego.

Sygnał taktujący przetwornikiem ADCCLK ma częstotliwość maksymalną 14 MHz. Rdzeń STM32F100 może być taktowany z częstotliwością nie większą niż 24 MHz i z taką maksymalną częstotliwością pracują obie magistrale APB1 i APB2. Sygnał taktujący przetwornikiem jest brany z szyny APB2 i dzielony przez preskaler o programowanym podziale przez 2, 4 6 i 8. Aby uzyskać maksymalną częstotliwość taktowania wybieramy podział przez 2 i otrzymujemy ADCCLK wynoszące 12 MHz. Znacząc częstotliwość taktowania można określić jak długo będzie trwała konwersja: $T = \text{sampling time} + 12,5 \times \text{TADCCLK}$ (TADCCLK – okres sygnału taktowania przetwornika). Czas $12,5 \times \text{TADCCLK}$ wynika z przetwarzania 12-bitowego przy metodzie sukcesywnej aproksymacji. Czas *sampling time* jest programowany i może mieć jedną z wartości: 1,5; 7,5; 13,5; 28,5; 41,5; 55,5; 71,5 lub 239,5 $\times \text{TADCCLK}$.

Moduł przetwornika trzeba przed użyciem skonfigurować, zapewnić taktowanie i włączyć. Każdy z pomiarów będzie wyzwolony programowo. Dlatego trzeba wyłączyć pomiar ciągły i wyzwalać zewnętrznie. Czas trwania pojedynczej konwersji zależy od częstotliwości taktowania modułu przetwornika.

Przetwornik może korzystać z 16 wejść analogowych przełączanych multiplexerem analogowym. My będziemy mierzyli napięcie wyjściowe zasilacza i napięcie z konwertera prąd/napięcie (prąd wyjściowy). Napięcie wyjściowe jest mierzone w kanale 0 (linia PA0), a napięcie z konwertera prąd/napięcie w kanale 1 (PA1). Linie PA0 i PA1 trzeba skonfigurować jako wejścia analogowe (**listing 3**).

Aby przetwornik mógł w ogóle pracować, trzeba wybrać źródło jego sygnału zegarowego i programowo włączyć moduł. Ponieważ częstotliwość taktowania rdzenia ustalono na 24 MHz, to zegar taktujący przetwornik musi być podzielony przez 2. Pokazano to na **listingu 4**.

Moduł ADC jest rozbudowany i może pracować w różnych trybach. Nas będzie interesowała praca w trybie *regular* z pomiarem w 2 kanałach. Jest możliwe włączenie automatycznego przemiatania kanałów pomiarowych. Wtedy po zakończeniu obsługi jednego kanału automatycznie jest inicjowany kolejny. Jednak jest z tym problem, bo w trybie *regular* wynik konwersji jest zapisywany do jednego rejestru. Użytkownik, gdy jest włączone skanowanie nie wie, z którego kanału odczytuje pomiar. Tę niedogodność można wyeliminować stosując przesyłanie wyników poszczególnych pomiarów przez

Listing 3. Konfigurowanie PA0 i PA1 jako wejść analogowych

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN; //wejście analogowe
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOA, &GPIO_InitStructure);
```

Listing 4. Konfiguracja taktowania i włączenie przetwornika

```
RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1, ENABLE); //włącz taktowanie ADC1
RCC_ADCCLKConfig(RCC_PCLK2_Div2); //taktowanie przetwornika =12MHz
ADC_Cmd(ADC1, ENABLE);
```

Listing 5. Konfigurowanie przetwornika ADC

```
void AnalogInit(void)
{
//struktura z konfiguracją ADC
ADC_InitTypeDef ADC_InitStructure;
//praca niezależna
ADC_InitStructure.ADC_Mode = ADC_Mode_Independent;
//skanowanie kanałów niepotrzebne
ADC_InitStructure.ADC_ScanConvMode = DISABLE;
//pomiar w trybie jednokrotnym
ADC_InitStructure.ADC_ContinuousConvMode = DISABLE;
//brak wyzwalań zewnętrznych
ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None;
//wyrównanie danych do prawej - 12 młodszych bitów znaczących
ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
//liczba używanych kanałów =2
ADC_InitStructure.ADC_NbrOfChannel = 2;
ADC_Init(ADC1, &ADC_InitStructure);
}
```

Listing 6. Kalibracja przetwornika ADC1

```
//zerowanie rejestrów kalibracyjnych ADC1
ADC_ResetCalibration(ADC1);
//oczekiwanie na restart
while(ADC_GetResetCalibrationStatus(ADC1));
//kalibracja ADC1
ADC_StartCalibration(ADC1);
while(ADC_GetCalibrationStatus(ADC1));
```

Listing 7. Główna część procedury inicjalizacyjnej

```
GPIO_LcdInit(); //inicjalizacja portów LCD
GPIO_SpiInit(); //inicjalizacja portów Spi
GPIO_ImpInit(); //inicjalizacja portów enkodera
LcdInit(); //inicjalizacja wyświetlacza LCD
AnalogInit(); //inicjalizacja ADC
DACInit(); //inicjalizacja DCA
RCC_ADCCLKConfig(RCC_PCLK2_Div2);
ADC_Cmd(ADC1, ENABLE);
ADC_Cmd(ADC2, ENABLE);
ZasilaczInit(); //inicjalizacja nastaw zasilacza
```

Listing 8. Inicjalizacja sterownika zasilacza

```
void ZasilaczInit(void){
if (EERead(ADDR_Init) != 0x49){
EEWrite(100, ADDR_Vo_H);
EEWrite(0, ADDR_Vo_L);
//ustawienie Limit_Iwój
Io = 3732;
EEWrite(Io >> 8, ADDR_In_H);
EEWrite(Io, ADDR_In_L);
//ustawienie limit_Uwój
EEWrite(Io >> 8, ADDR_Ul_H);
EEWrite(Io, ADDR_Ul_L);
//numer funkcji w menu funkcyjnym
EEWrite(0, ADDR_Nr_Fun);
EEWrite(0x49, ADDR_Init);
}
//odczytanie i ustawienie napięcia wyjściowego i zabezpieczenia prądowego
Vo = EERead(ADDR_Vo_H);
Vo <<= 8;
Vo |= EERead(ADDR_Vo_L);
Io = EERead(ADDR_Io_H);
Io <<= 8;
Io |= EERead(ADDR_Io_L);
DAC_SetChannel1Data(DAC_Align_12b_R, Vo);
DAC_SetChannel2Data(DAC_Align_12b_R, Io);
//prog odciecia prądowego
Iover = EERead(ADDR_In_H);
Iover <<= 8;
Iover |= EERead(ADDR_In_L);
//prog odciecia napięciowego
Uover = EERead(ADDR_Ul_H);
Uover <<= 8;
Uover |= EERead(ADDR_Ul_L);
}
```

kanał DMA do tablicy wyników umieszczonej w pamięci RAM. Ja postanowiłem zastosować typową metodę polegającą na konfiguracji jednego kanału pomiarowego przed pomiarem, wykonaniu i zapisaniu pomiaru, a następnie skonfigurowaniu następnego

kanału i wykonaniu pomiaru w tym kanale. Do tego celu zdefiniowałem funkcję konfiguracyjną *AnalogInit* (**listing 5**) konfigurującą przetwornik do pracy niezależnej i bez wyzwalań zewnętrznych. Pomiar jest wykonywany w pojedynczym kanale, w trybie

Listing 9. Ustawianie napięcia wyjściowego VO

```

char SetVo(unsigned char kod)
{
    short Vo, step;
    char dir=0;

    step=VoSTEPD;
    if(kod==KOD_IMP_UP||kod==KOD_IMP_DWN)
    {
        Vo=EERead(ADDR_Vo_H);
        Vo<<=8;
        Vo|=EERead(ADDR_Vo_L);
        if(kod==KOD_IMP_UP)
        {
            dir=1;
            if(ST==1) step=VoSTEPZ;
            Vo=Vo+step;
            if(Vo>2988) Vo=2988;
        }
        if(kod==KOD_IMP_DWN)
        {
            dir=0;
            if(ST==1) step=VoSTEPZ;
            Vo=Vo-step;
            if(Vo<0) Vo=0;
        }
        if(dir==0&&Vo<=1482) //jedziemy w dol
        {
            GPIO_WriteBit(GPIOB, GPIO_Pin_6, Bit_SET);
            PosLcd(14,1);
            DispLcd("LO");
        }
        if(dir==1&&Vo>=1506) //jedziemy w gore
        {
            GPIO_WriteBit(GPIOB, GPIO_Pin_6, Bit_RESET);
            PosLcd(14,1);
            DispLcd("HI");
        }
        if(Vo>Uover)
        {
            CorrOverVoltage();
            PosLcd(11,1);
            DispLcd("U>");
            return(-1);
        }
        else PosLcd(11,1);
        DispLcd(" ");
        EEWrite(Vo>>8, ADDR_Vo_H);
        EEWrite(Vo, ADDR_Vo_L);
    }
    //korekcja nieliniowości bufora DAC
    if(Vo<=1116) Vo=Vo-CORL;
    if(Vo>1116&&Vo<=1736) Vo=Vo-CORM;
    DAC_SetChannel1Data(DAC_Align_12b_R, Vo);
    return(0);
}

```

jednokrotnym (skanowanie kanałów jest wyłączone). Taktowany i włączony moduł ADC można skalibrować. Wtedy według zapewnienia producenta poprawia się dokładność konwersji. Sposób uruchomienia kalibracji pokazano na **listingu 6**.

Konfiguracja modułów przetworników to zasadniczy element inicjalizacji sterow-

nika zasilacza. Oprócz tej konfiguracji trzeba zainicjować porty GPIO do obsługi wyświetlacza LCD i sterowania przekaźnikami (wyjściowe) oraz do obsługi impulsatora i styków wykorzystywanych w interfejsie użytkownika (wejściowe). Inicjalizacji wymagają też linie portów używanych do komunikacji z pamięcią EEPROM. Interfejs komunikacyj-

ny SPI jest tutaj emulowany programowo. Główną procedurę inicjalizacyjną pokazano na **listingu 7**.

Jak wiemy, napięcia referencyjne wytwarzane przez C/A określają wartość napięcia wyjściowego i ograniczenia prądowego. W klasycznych konstrukcjach te nastawy są określane przez ustawienie położenia potencjometrów. Po wyłączeniu i ponownym włączeniu zasilacza nie powinny się zmienić, jeżeli potencjometry nie zostały przestawione. W sterowaniu cyfrowym zwykle jest potrzebne, aby po ponownym włączeniu zasilacz dostarczał takiego samego napięcia (i miał ten sam poziom ograniczenia) jak przed wyłączeniem.

W trakcie kręcenia impulsatorem zmieniane wartości są zapisywane do pamięci EEPROM. Po włączeniu zasilania sterownik odczytuje zapamiętane wartości i wysyła je do przetworników C/A. Kiedy sterownik jest włączony po raz pierwszy, to w pamięci EEPROM są zapisane liczby 0xFF. Dlatego w takim wypadku jest konieczne nadanie nastawom pewnych wartości domyślnych. Procedura inicjalizacyjna sprawdza czy w komórce pamięci o adresie *ADDR_Init* jest zapisana wartość 0x49 i jeżeli nie, to inicjuje obszar EEPROM odpowiedzialny za nastawy, a potem zapisuje wartość 0x49 pod adresem *ADDR_Init*. Po tej operacji mamy wstępnie zainicjowane ustawienia i w czasie następnego włączenia ponowna wstępna inicjacja nie zostanie wykonana. Na **listingu 8** pokazano procedurę *ZasilaczInit()*.

Napięcie wyjściowe ustawia się za pomocą funkcji *SetVo* pokazanej na **listingu 9**. Jej argumentem jest kod obrotu impulsatora. Na początku z pamięci EEPROM jest odczytywana ostatnio ustawiona wartość napięcia. Potem, w zależności od kodu, ta wartość jest zwiększana lub zmniejszana. Krok zmiany zależy od tego czy styk zwierany przez wciśnięcie gałki impulsatora jest wciśnięty (krok 0,1 V), czy nie (krok 1 V). Przy zwiększaniu ograniczeniem jest wartość 2988 odpowiadająca napięciu +2,4 V na wyjściu C/A, czyli +24 V na wyjściu zasilacza. Przy każdym zwiększeniu zmiennej *Vo* program sprawdza czy napięcie wyjściowe nie jest mniejsze od 11,9 V. Jeżeli tak, to załącza przekaźnik RE2 i główne napięcie zasilające jest pobierane z odczepu transformatora. W moim zasilaczu napięcie na C2 ma wtedy wartość ok. +16 V. Jeżeli modyfikowana wartość *Vo* jest zwiększana, to program sprawdza czy po modyfikacji nie jest większa niż 12,1 V i ewentualnie wyłącza przekaźnik RE2. Wtedy zasilacz jest zasilany pełnym napięciem – na C2 występuje ok. 32 V. Po ustawieniu nowej wartości program sprawdza czy ustalone napięcie nie jest większe niż wartość zapisana w zmiennej *Uover*. Jeżeli tak, to jest wywoływana funkcja korekcji *CorrOverVoltage*. Zmienna *Uover* jest ustawiana przez funkcję

Listing 10. Pomiar i wyświetlenie napięcie wyjściowego

```

void PomV(char x, char y)
{
    unsigned long pomiar;
    char txt[8];
    char i;

    PosLcd(x,y);
    DispLcd("Uwy=");
    ADC_RegularChannelConfig(ADC1, ADC_Channel_0, 1, ADC_SampleTime_28Cycles5);
    //wyzwolenie pojedynczego pomiaru
    ADC_SoftwareStartConvCmd(ADC1, ENABLE);
    //odczekaj na zakończenie konwersji
    while (!ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC));
    //pobierz zmierzona wartość
    pomiar = ADC_GetConversionValue(ADC1);
    //przelicz wartość
    pomiar = (pomiar * 8057)/1000;
    sprintf((char *)txt, "%d,%03d V\0", pomiar / 1000, pomiar % 1000);
    if(txt[1]!='.'){
        txt[3]='V';
        txt[4]=' ';
        for(i=0;i<6;i++) WriteRd(txt[i]);
    }
    else
    {
        txt[4]='V';
        for(i=0;i<5;i++)
            WriteRd(txt[i]);
    }
}

```


Overvoltage. Zmodyfikowana wartość V_0 jest wysyłana do przetwornika C/A przez funkcję `DAC_SETChannel1Data`, a potem ponownie zapisywana w EEPROM.

W czasie konfigurowania przetwornika C/A do jego wyjście jest programowo dołączany wewnętrzny bufor. Wyjście przetwornika C/A ma stosunkowo dużą rezystancję i dlatego jego sygnał napięciowy jest podatny na zaburzenia. Bufor powoduje, że rezystancja wyjściowa jest mała, ale jego włączenie ma wady. Do podstawowych należy nieliniowość charakterystyki. Krok 1 V jest uzyskiwany przez dodawanie do zmiennej V_0 stałej 124 (dziesiątne). Jeżeli ustawimy prawidłowo napięcie na początku zakresu na 1 V, to przy liniowym przyroście wartości V_0 , dla teoretycznego napięcia +24 V, jest ustawiane przez C/A napięcie +24,15 V. Aby zmniejszyć ten błąd nastawy trzeba by było zastosować zewnętrzny bufor ze wzmacniaczem operacyjnym, a właściwie 2 bufor: dla układów regulacji napięcia wyjściowego i dla obwodów ograniczenia prądowego. Spowodowałoby to spore rozbudowanie układu. Ponieważ w naszym zasilaczu liniowość ustalenia napięcia wyjściowego nie ma aż tak dużego znaczenia, to zastosowałem korekcję programową w 2 punktach zakresu regulacji.

Pomiar napięcia (**listing 10**) rozpoczyna się od ustawienia kanału pomiarowego. Dla przypomnienia: napięcie wyjściowe mierzymy (po podzieleniu przez 10 w za pomocą dzielnika rezystancyjnego) przetwornikiem A/C w kanale pomiarowym 0 (linia PA0). Kanał jest ustawiany funkcją biblioteczną `ADC_RegularChannelConfig`. Po ustawieniu kanału funkcja `ADC_SoftwareStartConvCmd` wyzwala programowo jeden pomiar. Konwersja jest zakończona, gdy przetwornik ustawia flagę `ADC_FLAG_EOC`. Zawartość rejestru konwersji odczytuje funkcja `ADC_GETConversionValue`. Nasze napięcie referencyjne wynosi +3,3 V, a rozdzielczość pomiaru 12 bitów. Po podzieleniu pełnego zakresu przetwornika 3,3V przez $2^{12}=4096$ otrzymamy 0,0008056640625 V. Zmiana wartości na najmłodszym bicie wyniku pomiaru odpowiada zmianie napięcia o ok. 0,0008057 V, więc aby przekonwertować wartość odczytaną z przetwornika na napięcie w woltach trzeba ją pomnożyć przez 0,0008057 V.

Mnożenie zmiennoprzecinkowe jest oczywiście wykonalne, ale lepiej jest mnożyć stałoprzecinkowo. Wykorzystuje się tutaj trik polegający na mnożeniu wyniku konwersji przez 8057, a potem podzieleniu wyniku przez 1000, pamiętając o odpowiedniej długości zmiennych, na których operujemy i o przecinku. Po tych operacjach wynik jest umieszczony w zmiennej pomiar. Aby wynik wyświetlić w postaci znaków ASCII na wyświetlaczu LCD trzeba go przekonwertować. Do tego celu doskonale nadaje funkcja `sprintf` umożliwiająca konwertowanie zawartości

Listing 11. Pomiar prądu i zabezpieczenie overcurrent

```
char PomI(char x, char y)
{
    unsigned long pomiar;
    char txt[8];
    char i;

    PosLcd(x,y);
    DispLcd("Iwy=");
    ADC_RegularChannelConfig(ADC1, ADC_Channel_1, 1, ADC_SampleTime_28Cycles5);
    //wyzwolenie pojedynczego pomiaru
    ADC_SoftwareStartConvCmd(ADC1, ENABLE);
    //odczeka na zakończenie konwersji
    while (!ADC_GetFlagStatus(ADC1,ADC_FLAG_EOC));
    //pobierz zmierzona wartość
    pomiar = ADC_GetConversionValue(ADC1);
    if (pomiar>Iover)
    {
        Delay(20); //opoznienie 20 ms
        //wyzwolenie pojedynczego pomiaru - powtorny pomiar po 20 ms
        ADC_SoftwareStartConvCmd(ADC1, ENABLE);
        //odczeka na zakończenie konwersji
        while (!ADC_GetFlagStatus(ADC1,ADC_FLAG_EOC));
        //pobierz zmierzona wartosc
        pomiar = ADC_GetConversionValue(ADC1);
        if (pomiar>Iover)
        {
            GPIO_WriteBit(GPIOB, GPIO_Pin_5, Bit_SET); odłącz wyjście
            return(-1);
        }
    }
    if(++pomi_disp==10)
    {
        pomi_disp=0;
        //przelicz wartosc wyrazona jako calkowita, 12-bit na rzeczywista
        pomiar = pomiar * 8057/10000;
        //pomiar= pomiar*10;
        sprintf((char *)txt, "%d,%03d A\0", pomiar / 1000, pomiar % 1000);
        if(txt[1]!='.')
        {
            txt[4]='A';
            txt[5]=' ';
            for(i=0;i<6;i++) WriteRd(txt[i]);
        }
        else
        {
            txt[5]='A';
            for(i=0;i<6;i++) WriteRd(txt[i]);
        }
    }
    return(0);
}
```

Listing 12. Pętla główna sterownika zasilacza

```
while(1)
{
    PomV(1,1);
    SetVo(GetEncoder()); //ustawienie napięcia wyjściowego
    if(PomI(1,2)==-1) //pomiar napięcia wyjściowego
    {
        DispLcd("I> ");
        PomV(1,1);
        while(KAS)
            Delay(20);
        while(!KAS) Delay(50);
        GPIO_WriteBit(GPIOB, GPIO_Pin_5, Bit_RESET); //załącz wyjście
    }
    if(KAS==0) MenuSetup(); //wyzwolenie menu funkcyjnego
}
```

zmiennej do postaci znakowej i zapisująca wynik konwersji w buforze umieszczonym w pamięci RAM. Po konwersji pozostaje tylko wyświetlić zawartość bufora `txt[]`.

Bardzo podobnie jest mierzony i wyświetlany prąd. Napięcie z konwertera prąd/napięcie mierzymy w kanale1 i dlatego przed pomiarem trzeba go sobie ustawić za pomocą funkcji `ADC_RegularChannelConfig(ADC1, ADC_Channel_1, 1, ADC_SampleTime_28Cycles5)`;

Mnożenie, dzielenie, konwersja przy pomocy `sprintf`, oraz wyświetlanie przebiega praktycznie tak samo jak w funkcji `PomV`. W funkcji pomiaru prądu jest też zaimplementowane zabezpieczenie nadprądowe. Jeżeli po pomiarze wartość prądu jest wyższa niż zapisane w zmiennej `Iover`, to program odczeka 20msek i ponawia pomiar prądu. Jeżeli dalej prąd jest większy niż `Iover`, to

funkcja pomiaru odłącza wyjście zasilacza od obciążenia przełączając przełącznik RE1 (na linii PB5 stan wysoki). Ponowne załączenie wyjścia zasilacza do obciążenia następuje w pętli głównej po naciśnięciu przycisku KAS. Przed jego naciśnięciem trzeba odłączyć obciążenie od zacisków wyjściowych zasilacza, bo w przeciwnym przypadku sterownik załączy obciążenie, zmierzy prąd i ponownie je wyłączy.

Pętla główna i menu funkcyjne

Na początku pętli głównej jest mierzona napięcie wyjściowe (funkcja `PomV`). Następnie funkcja `SetVo` odczytuje stan impulsatora (wywołanie `GetEncoder` jako argument) i jeżeli został on obrócony, to zmienia się wartość napięcia, tak jak to zostało pokazano na list. 9. Po ustawieniu napięcia funkcja `PomI` mierzy i wyświetla prąd obciążenia

Listing 13. Ustawienie progu zabezpieczenia prądowego

```

//ustawienie ograniczenia prądowego
void SetLimitI(void) {
    char txt[8]={"          "};
    char i, kod, step;
    long pomiar, Io;

    ClrLcd();
    PosLcd(1,1);
    DispLcd("LimitI=");
    Io=EERead(ADDR_Io_H);
    Io<<=8;
    Io|=EERead(ADDR_Io_L);
    while(1)
    {
        step=VoSTEPD;
        kod=GetEncoder();
        if(kod==KOD_IMP_UP)
        {
            if(ST==1) step=VoSTEPZ;
            Io=Io+step;
            if(Io>3732) Io=3732;
        }
        if(kod==KOD_IMP_DWN)
        {
            if(ST==1) step=VoSTEPZ;
            Io=Io-step;
            if(Io<0) Io=0;
        }
        DAC_SetChannel2Data(DAC_Align_12b_R, Io); //ustawienie ograniczenia prądu
        PosLcd(8,1); //wyswietl wartosc ograniczenia
        pomiar=Io;
        pomiar = pomiar * 8057/10000; //przelicz wartosc
        sprintf((char *)txt, "%d,%03d A\0", pomiar / 1000, pomiar % 1000); //
konwersja BIN/ASCII
        if(txt[1]!='\0')
        {
            txt[4]='A';
            txt[5]=' ';
            for(i=0;i<6;i++) WriteRd(txt[i]);
        }
        else
        {
            txt[5]='A';
            for(i=0;i<6;i++) WriteRd(txt[i]);
        }
        if(!KAS) break;
    }
    Delay(20);
    while(KAS);
    EEWrite(Io>>8, ADDR_Io_H);
    EEWrite(Io, ADDR_Io_L);
}

```

Listing 14. Korygowanie overvoltage

```

void CorrOverVoltage(void)
{
    //napiecie wyjsciowe rowne Uover
    DAC_SetChannel1Data(DAC_Align_12b_R, Uover);
    //zapisz Vo=Uover
    EEWrite(Uover>>8, ADDR_Vo_H);
    EEWrite(Uover, ADDR_Vo_L);
}

```

nia. Jeżeli prąd jest wyższy niż ustawiony przez funkcję *Overcurrent* (zmienna *Iout*), to zostanie odłączone wyjście za pomocą styków przekaźnika RE1, natomiast funkcja *PomI* zwróci -1. Na wyświetlaczu pojawi się komunikat „Iwy=I>”. Po naciśnięciu KAS przekaźnik jest załączany, a program powraca do pętli głównej.

Jeżeli w program jest w pętli głównej i nie jest wymagane wyłączenie zadziałania zabezpieczenia przed przeciążeniem, to naciśnięcie przycisku KAS powoduje wejście do menu funkcyjnego. Funkcje w menu wybiera się przez kręcenie ośką impulsatora, a wybór zatwierdza przyciśnięciem KAS. Po zakończeniu działania wybranej funkcji, program wraca do pętli głównej. W obecnej wersji oprogramowania dostępne są 3 funk-

- ustawienie progu zadziałania ograniczenia prądowego,
- ustawienie maksymalnego napięcia wyjściowego (Overvoltage),

- ustawienie progu zabezpieczenia nadprądowego.

Funkcja ustawienia progu ograniczenia prądowego

Ograniczenie prądowe działa na zasadzie porównywania napięcia z konwertera prąd/napięcie z napięciem referencyjnym. Napięcie referencyjne ograniczenia prądowego jest generowane przez drugi przetwornik C/A, pierwszy służy do ustawiania napięcia wyjściowego. Zadaniem funkcji ustalającej próg ograniczenia prądowego jest odczytanie z pamięci EEPROM zapamiętanej wartości, umożliwienie jej edycji, wysłanie do modułu przetwornika C/A i po zakończeniu regulacji – ponowne zapisanie do pamięci EEPROM. Procedurę *SetLimitI* realizująca to zadanie pokazano na **listingu 13**.

Funkcja Overvoltage

Postanowiłem wyposażyć zasilacz w moim zdaniem bardzo przydatną funkcję

ograniczania napięcia wyjściowego. Jej działanie polega na ustawieniu konkretnej, progowej wartości napięcia zasilania, powyżej której nie będzie można zwiększyć napięcia wyjściowego. Załóżmy, że chcemy zasilać układ zbudowany z układów TTL i mikrokontrolera. W takim wypadku napięcie nie może być wyższe niż +5,25 V. Jeżeli w trakcie pracy przez pomyłkę zwiększymy napięcie zasilania, to zasilany układ może zostać uszkodzony. Żeby się przed tym zabezpieczyć ustawiamy w funkcji *Overvoltage* wartość +5 V, która zostanie zapisana w zmiennej globalnej *Uover* i w pamięci EEPROM. Jeżeli w trakcie regulowania napięcia wyjściowego zostanie wykryty warunek $V_o > U_{over}$, to zostanie wywołana funkcja *CorrOverVoltage*, która ustawi na przetworniku C/A napięcie równoważne *Uover* i zapisze w pamięci EEPROM, w obszarze zapamiętywania V_o , wartość *Uover* (**listing 14**). Wyłączenie działania *Uover* następuje po ustawieniu wartości napięcia większej od +24 V.

Funkcja ustawiania ograniczenia nadprądowego overcurrent

Ograniczenie prądu wyjściowego ma wiele zalet, ale są sytuacje, w których potrzeba innego zabezpieczenia – działającego w taki sposób, że po przekroczeniu określonego prądu automatycznie odłącza wyjście zasilacza od obciążenia na przykład za pomocą przekaźnika. Oprócz progu prądowego powinno się też ustawiać czas zwłoki do zadziałania. Jeżeli zasilamy układ, który pobiera 100 mA i ustawimy zabezpieczenie na 150 mA, ale w układzie zasilania ma spory kondensator elektrolityczny, to w czasie jego ładowania prąd będzie wyższy niż próg zabezpieczenia. Jeżeli wyłączenie nastąpi natychmiast po wykryciu przekroczenia, to będzie trudno taki układ zasilic z zabezpieczeniem nadprądowym. Pozostaje ustawić wyżej próg lub odmierzyć pewien czas zwłoki i po tym czasie ponownie skontrolować prąd. W obu wypadkach wrażliwe układy mogą być narażone na uszkodzenie, bo dopuszczamy w określonych warunkach przepływ dużo większego prądu, niż to wynika z potrzeb zabezpieczenia układu. Jak widać, nie ma idealnego zabezpieczenia przez wzrostem natężenia prądu i trzeba wybrać jakiś kompromis. W tej wersji programu nie ma możliwości ustawienia czasu zwłoki. Jest on ustawiony na 20 ms.

Próg zabezpieczenia jest ustawiany podobnie, jak w wypadku ustawiania napięcia *overvoltage*. Wartość zabezpieczenia jest zapisywana do zmiennej *Uover* i do pamięci EEPROM. Warunek przekroczenia sprawdza się przy każdym pomiarze w procedurze *PomI*. Jeżeli jest spełniony, to funkcja wyłącza wyjście i zwraca -1. Dokładnie

zostało to opisane przy omawianiu funkcji pomiaru prądu *PomI*.

Montaż i uruchomienie zasilacza

Zasilacza jest podzielony na 2 płytki: płytkę układów analogowych (**rysunek 7**) i płytkę sterownika (**rysunek 8**).

Wykaz elementów
Płytkę układów analogowych

Rezystory: (metalizowane 5%/0,6 W)
R23: 47 Ω
R14, R16, R20: 100 Ω
R21: 470 Ω
R4, R5, R8, R17, R18: 1 kΩ
R2, R12: 9,1 kΩ
R3, R15: 1 MΩ
R19: 0,1 Ω/5 W (5%)
R9: 1 kΩ (pot. wieloobr.)
R11: 2 kΩ (pot. wieloobr.)
R10: 22 kΩ (pot. wieloobr.)

Kondensatory:
C4, C15, C16: 4,7 nF/100 V (foliowy)
C3, C6, C7, C11, C12, C18, C19, C23, C24, C28: 100 nF/63 V (SMD 1206)
C1, C5, C8...C10, C13...C16, C20...C22, C25, C27: 10 μF/50 V (elektrolit.)
C26: 1000 μF/50 V (elektrolit.)
C2: 68000 μF/50 V (elektrolit.)

Półprzewodniki:
D1, D2, D4, D5: diody prostownicze 5 A/100 V
D3, D6, D7: 1N4148 (BAV21)
D9: dioda Zenera 12 V/1,3W
D10: mostek prostowniczy 1 A/1000 V, okrągły
D11, D8: dioda Zenera 15 V/1,3 W
T1: IRF540N (FET N)
T2...T5: BD139
U1, U3, U4: μA741IN lub podobne w obudowie DIP8
U2: ICL7660 (DIP8)
U5: 7805

Inne:
RE1, RE2 Schrack RT424024 lub odpowiedniki
Płytkę drukowaną
Podwójna listwa goldpinów raster 2,54 mm

Płytkę sterownika

Rezystory:
R1...R6, R8, R9: 1 kΩ (SMD 1206)
Potencjometr 47 kΩ

Kondensatory:
C9, C10: 27 pF (SMD 1206)
C1, C3, C4, C7, C8, C11...C13, C15: 100 nF (SMD 0805)
C2, C5, C6, C16: 10 μF/10 V (SMD 3216)

Półprzewodniki:
U1: LM1117MP-3.3 (SOT-223N)
U2: STM32F100RBT6B (zaprogramowany)
U5 25LC256 I/SN (SOIC8, 150 mils)

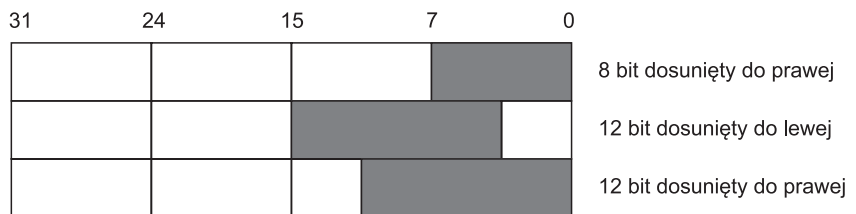
Inne:
Wyświetlacz LCD-AC-1602F-BIW W1B-E6 C PBF (2 × 16 znaków)
Impulsator ze stykiem EC-11SP-1-1-1-15KQ
Podwójna listwa goldpinów raster 2,54mm
Transformator – opis w tekście
Kabel 10-przewodowy z 2 wtykami IDC10
Radiator
Obudowa
Wyłącznik sieciowy
Gniazdo bezpiecznikowe.

Uruchomienie płytki analogowej. Prawie wszystkie elementy na płytce układów analogowych są przeznaczone do montażu przewlekane. Tranzystor mocy T1 można wlutować bezpośrednio w punkty lutownicze na płytce, ale pamiętajmy o tym, że jako element regulacyjny musi on być umieszczony na radiatorze. Ja w punkty lutownicze przeznaczone na nóżki T1 wlutowałem listwę 3 goldpinów o rastrze 2,54 mm, a potem do tych goldpinów przywlutowałem krótkie odcinki przewodów. Tranzystor można wtedy w miarę swobodnie umieścić na radiatorze i połączyć go z płytką przewodami pamiętając jednak o tym, by przewody były tak krótkie, jak to możliwe. **Jeżeli tranzystor zostanie przykręcony bezpośrednio bez przekładki i tulejek izolacyjnych, to na radiatorze będzie występowało napięcie równe napięciu na kondensatorze C2 – w praktyce ponad 30 V i dlatego dobrym rozwiązaniem jest odizolowanie tranzystora od radiatora.**

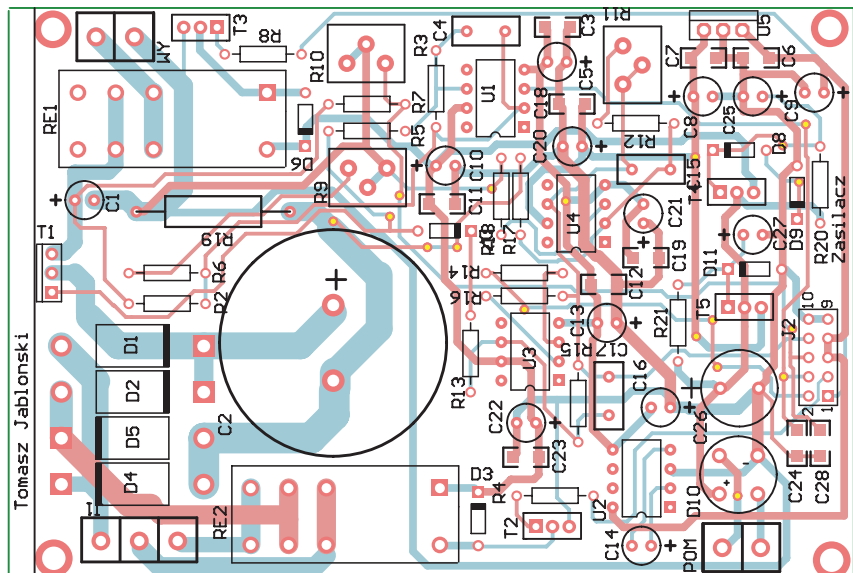
Po zmontowaniu i sprawdzeniu montażu dołączamy napięcie przemienne o wartości ok. 30...33 V i wydajności prądowej 200 mA do złącza POM i sprawdzamy poprawność napięć zasilających: Ust, +15 V, +5 V oraz -5 V z układu U2. Napięcie Ust w modelowym rozwiązaniu miało wartość +27 V, aby można było ustawić maksymalne napięcie wyjściowe zasilacza równe +24 V. W tym momencie płytkę sterownika powinna być bezwzględnie odłączona od płytki analogowej.

Teraz można dołączyć napięcie przemienne o wartości ok. 27 V do styków 1 i 2 złącza J1. Przy braku napięć sterujących na złączu J2, napięcie na wyjściu zasilacza powinno być równe 0 V. Bezwzględnie trzeba też sprawdzić czy nie ma wysokich napięć na stykach 1...4, 6 i 8 złącza sterującego J2. Przy prawidłowo zmontowanym układzie i sprawnych elementach na tych stykach powinno występować ok. 0 V. Zbyt wysokie napięcia mogą zniszczyć linie I/O mikrokontrolera.

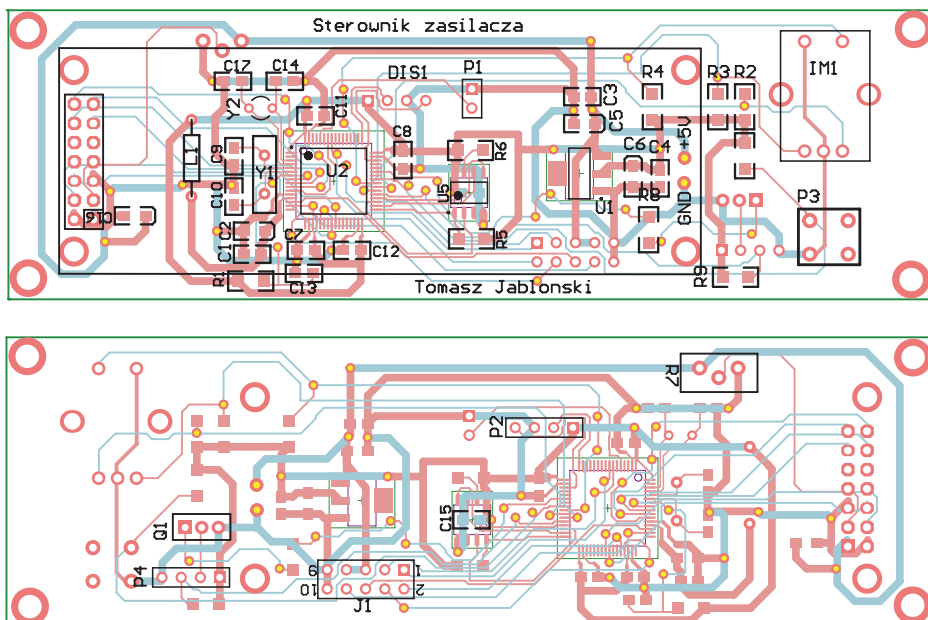
Tu dochodzimy do momentu, w którym trzeba sprawdzić działanie obwodów regulacji i pomiarów układów analogowych. Można to zrobić na 2 sposoby. Jeżeli mamy zmontowaną i wstępnie sprawdzoną płytkę sterownika, to można ją dołączyć i uruchamiać układ docelowy. Napiszę o tym dalej. Druga metoda to wykorzystanie 2 potencjometrów – jeden do regulowania napięcia wyjściowego, a drugi do regulowania poziomu zadziałania zabezpieczenia prądowego i stabilizatora +3,3 V. Ja przy uruchamianiu prototypu posłużyłem się najpierw tą metodą. Przy okazji można zobaczyć, jak można sterować i użytkownika zasilacz bez sterownika mikroprocesorowego. Wszystko wtedy będzie działać poprawnie, ale nie będą dostępne pomiary wykonywane przez sterownik. Ponieważ układ analogowy ma być „zestrojony” dla napięć referencyjnych o maksymalnej wartości +3,3 V (z DAC mikrokontrolera), to potencjometry będą dzieliły napięcie +3,3 V ze



Rysunek 6. Umieszczenie danych do konwersji



Rysunek 7. Schemat montażowy płytki układów analogowych



Rysunek 8. Schemat montażowy płytki sterownika

stabilizatora LDO o napięciu wyjściowym +3,3 V uzyskiwanym z napięcia +5 V, dostępnego na stykach 5 i 7 złącza J2. Żeby regulować poziom napięcia wyjściowego trzeba podać podzielone napięcie z suwaka potencjometru (liniowy 1...10 k Ω) na styk 2 (Reg) złącza J2. Napięcie na nim powinno zamieniać się w zakresie 0...2,4 V, a napięcie na wyjściu w zakresie 0...24 V.

Układ wymaga regulacji wykonywanej potencjometrem R9 umieszczonym na płytce układów analogowych. Ustawiamy dokładnie napięcie sterujące 1,00 V na wejściu sterującym Reg (doprowadzenie 4 złącza J2) i tak ustawiamy R9, aby napięcie na wyjściu zasilacza miało wartość 10,0 V.

Przy okazji trzeba dokładnie ustawić dzielnik napięcia wyjściowego (rezystory R2, R6, R9). Przy napięciu wyjściowym 10,0 V tak regulujemy potencjometrem R9, aby napięcie na styku 6 złącza J2 miało wartość równą 1,00 V. Od dokładności tej regulacji będzie zależała dokładność pomiaru napięcia wyjściowego przez sterownik zasilacza. Poprawność wykonania czynności regulacyjnych można sprawdzić przy innych napięciach wyjściowych, na przykład przy 5,0 V; 15,0 V i 20,0 V.

Układ do pomiaru prądu wymaga wzmocnienia około 10 \times . Oznacza to, że napięcie na wyjściu U4 ma być 10 razy większe niż spadek napięcia na rezystorze pomiarowym R19. Wzmocnienie konwertera I/U ustawiamy potencjometrem R11. Jako rezystor pomiarowy zastosowałem łątki do kupienia opornik 0,1 Ω /5 W w obudowie ceramicznej. Dokładność rezystancji tego elementu nie spełnia wymagań stawianym rezystorom boczników w klasycznych układach pomiarowych, ale tutaj możemy przez zmianę wzmocnienia skorygować rozrzut parametrów. W praktyce regulacja

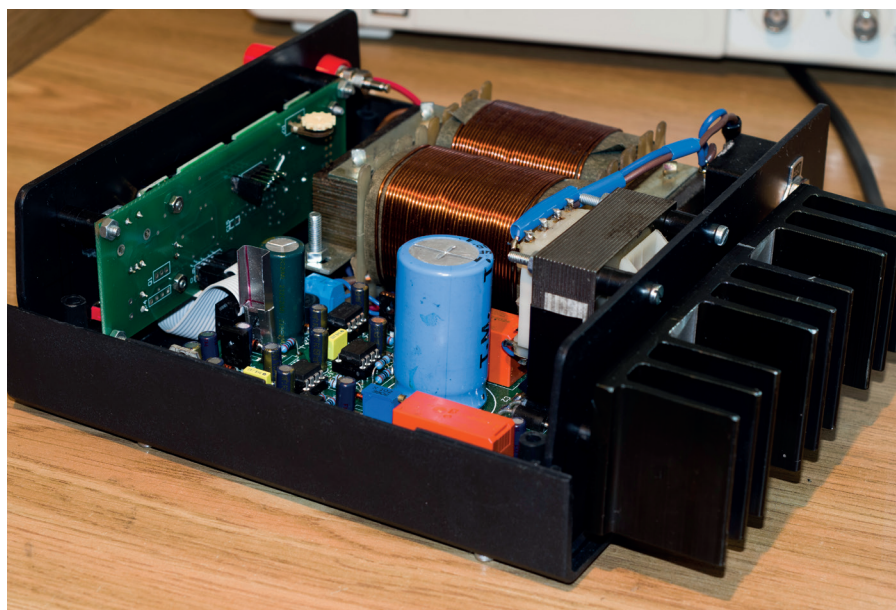
wygląda tak: obciążamy zasilacz prądem 1 A. W szereg z obciążeniem włączamy amperomierz, na przykład o zakresie pomiarowym 2 A. Jeżeli prąd mierzony ma wartość 0,95 A, to napięcie na wyjściu U4 powinno mieć wartość 0,95 V i tak regulujemy potencjometrem R11, aby tę wartość uzyskać. Dokładność regulacji zależy głównie od dokładności pomiaru prądu przez zewnętrzny amperomierz.

Prąd pobierany z zasilacza nie musi być mierzony bardzo dokładnie. Ten pomiar najczęściej jest orientacyjny, ale możemy go wykonać z tak dużą dokładnością, jak tylko się da. Jeżeli są jakieś problemy, to można zmierzyć spadek napięcia na rezystorze R19. Powinien być około 10 \times mniejszy od napięcia na wyjściu układu pomiarowego. Jeżeli tak nie jest, to trzeba sprawdzić rezystancję R19 i wartość elementów w otoczeniu wzmacniacza U4.

Układ zabezpieczenia prądowego wymaga drugiego napięcia referencyjnego uzyskiwanego z drugiego potencjometru. Ponieważ na wyjściu pomiaru prądu napięcie na wyjściu układu pomiarowego prądu zmienia się w zakresie 0...3 V (co odpowiada 0...3 A), to wartość ograniczenia prądowego będzie ustawiana napięciem referencyjnym w zakresie 0...3 V. Tu zależność jest również prosta: ustawienie napięcia referencyjnego o wartości 1 V powoduje działanie ograniczenia na poziomie 1 A. Sprawdzamy to w następujący sposób: obciążamy wyjście zasilacza prądem na przykład 0,5 A przy ograniczeniu ustawionym na 3 A. Potem przestawiamy ograniczenie na 0,5 A i napięcie na wyprowadzeniu 2 złącza J2 (Ilewel) jest równe 0,5 V. Przy próbie zwiększania napięcia wyjściowego, układ ograniczenia prądowego nie pozwala na to i napięcie pozostaje na tym samym poziomie aż do momentu zmiany poziomu ograniczenia lub zmiany rezystancji obciążenia.

Jak już wspomniałem, wersja zasilacza bez sterownika wymaga 2 potencjometrów i napięcia +3,3 V. Jeżeli decydujemy się na uproszczoną wersję, to albo układ U5 wymieniamy na stabilizator napięcia +3,3 V lub przeliczamy wzmocnienia układów dla napięć referencyjnych zmieniających się w zakresie 0...5 V. Po wykonaniu wszystkich regulacji moduł analogowy jest gotowy do dołączenia sterownika.

Sterownik mikroprocesorowy. Montaż sterownika jest nieco bardziej skomplikowany. Przede wszystkim trzeba przylutować mikrokontroler umieszczony w obudowie do montażu powierzchniowego. Płytkę jest tak pomyślana, że wyświetlacz LCD został umieszczony nad większością elementów sterownika. Jeżeli przylutujemy wyświetlacz za pomocą dwurzędowych goldpinów (jak w prototypie), to trzeba mieć pewność, że

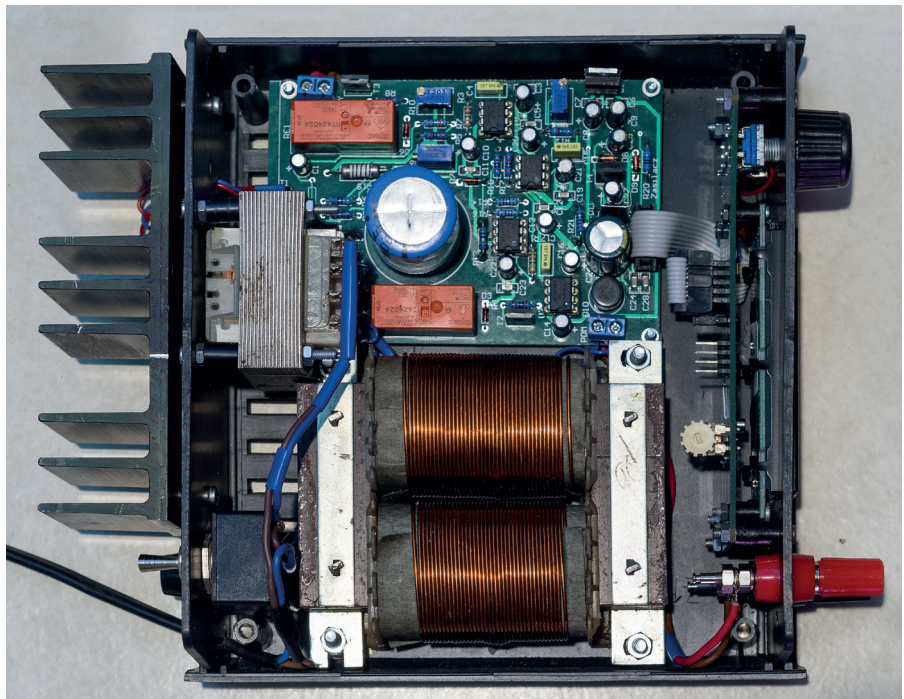


wszystkie luty zostały wykonane prawidłowo, bo potem nie ma możliwości poprawek bez bardzo trudnego demontażu połączeń wyświetlacza. Mniej ryzykowne będzie połączenie wyświetlacza z płytką sterownika za pomocą giętkich przewodów, aby można go było w każdej chwili odsunąć od płytki. **Tu uwaga – zastosowany wyświetlacz typu LCD-AC-1602F-BIW W1B-E6 C PBF ma na nóżce 1 doprowadzenie napięcia +5 V, a na nóżce 2 podłączenie GND zasilania i tak została zaprojektowana płytka drukowana.** Przy zastosowaniu innych wyświetlaczy z dużym prawdopodobieństwem może być odwrotnie. Trzeba to sprawdzić przed dołączeniem zasilania.

Po zmontowaniu całości trzeba sterownik zasilić podając napięcie +5 V na doprowadzenia 5, 7 (+5 V) i 9,10 (GND) złącza J1. Po sprawdzeniu poprawności napięcia +3,3 V można przejść do programowania mikrokontrolera.

Do programowania są używane linie SWCLK i SWDIO szeregowego interfejsu SWD dostępnego w programatorze/debugerze ST-Link2. Można tu użyć programatora ZL30ProgV2 lub wykorzystać moduł ST Link2 z dowolnej płytki STM32 Discovery. Ja użyłem programatora ST Link z modułu *STM32 Value Line Discovery*. Moduł jest podzielony na 2 części: niezależny programator /debuger STLink2 zasilany ze złącza USB i część z testowanym mikrokontrolerem z rodziny STM32. Domyślnie Discovery jest tak skonfigurowany, że linie interfejsu SWD są połączone z testowanym mikrokontrolerem, ale można to łatwo zmienić zworkami. Przesłanie zworek powoduje odłączenie linii interfejsu SWD od mikrokontrolera na płytce i połączenie ich z wyprowadzeniami złącza dla programatora – dokładne informacje można znaleźć w dokumentacji modułu. Sam proces programowania wymaga programu sterującego, w którym można plik wczytać i wysłać przez *ST Link* do mikrokontrolera. Ja użyłem pakietu *µVision4*. Był to naturalny wybór, bo w tym środowisku został napisany i uruchomiony program sterujący. Jednak by w nim programować pamięć Flash mikrokontrolera, trzeba mieć otwarty projekt z plikami źródłowymi. Jeżeli mamy do dyspozycji tylko plik wynikowy z rozszerzeniem HEX, to dobrym wyborem jest użycie programu *ST Link Utility*.

Plik do zaprogramowania jest standardowo otwierany poleceniem *File -> Open*. Programowanie wykonuje się poleceniem z menu *Target -> Program*. W tym menu jest też zakładka ustawiająca tryb pracy in-



Fotografia 9. Wnętrze zasilacza prototypowego

terfejsu *ST Link* – musimy tu ustawić opcję SWD.

Kiedy pamięć Flash mikrokontrolera jest już zaprogramowana, można przystąpić do testowania sterownika. Po włączeniu zasilania na ekranie wyświetlacza powinny zostać wyświetlone $U_{wy}=0,0$ V (w górnej linijce), $I_{wy}=0,00$ A (w dolnej linijce) przy zwartych do masy wejściach pomiarowych U_{wy} (pin 6 złącza J1) i I_{wy} (pin 8 złącza J1). Na te wejścia można podawać napięcia z zakresu 0...+3,0 V. Na wyświetlaczu powinny zmieniać się wartości pomiarów zgodnie z zależnościami:

- napięcie wskazywane:
 $U_{out}=U(U_{wy})\times 10$,
- prąd wskazywany: $I_{out}=U(I_{wy})$.

Jeżeli tak jest, to układy pomiarowe działają poprawnie. W następnym kroku testujemy działanie ustawiania napięcia wyjściowego. Kręcimy ośką potencjometru i mierzymy napięcie na pinie 2 złącza J1. Na wyświetlaczu nic się nie zmienia, bo w tym momencie na wejście pomiarowe U_{wy} nie jest podawane napięcie z modułu analogowego. Ale jeżeli zewrzymy wyjście przetwornika C/A (pin2 złącza J1) z wejściem przetwornika A/C (pin4 złącza J1), to zdawane napięcie podawane przez C/A jest mierzone przez A/C i na wyświetlaczu pojawia się wartość ustawionego napięcia zgodnie ze znaną już regułą $U_{out}=U_{Reg}\times 10$. Napięcie powinno się zmieniać z krokiem 1 V, jeśli ośka impulsatora nie jest przyciśnięta i 0,1 V, jeśli jest przyciśnięta.

Podobnie jest sprawdzane ustawianie prądu ograniczenia prądowego. Żeby sobie ułatwić zadanie, trzeba zewrzeć piny 2 (*Ilevel*) i 8 (*Iwy*) złącza J1. Wartość zabezpieczenia prądowego ustawia się po wejściu do funkcji *Set limit I*. Podobnie jak przy regulacji napięcia, obrót ośki powoduje zmianę ustawionego z krokiem 1 A, a po przyciśnięciu osi impulsatora z krokiem co 0,1 A. Ustawiony prąd będzie również widoczny po wyjściu z funkcji *Set limit I*, bo zwarte są piny 2 i 8 złącza J1. Analogicznie są ustawiane progi napięcia i prądu w funkcjach *Set Overvoltage* i *Set Overcurrent*, jednak działanie tych funkcji będzie można przetestować po połączeniu sterownika z płytką układów analogowych. Po tych testach trzeba rozewrzeć piny 4-6 i 2-8 złącza J1 i płytka sterownika jest gotowa do włączenia do układu zasilacza.

Testowanie kompletnego zasilacza

Po wstępnym przetestowaniu płytki układów analogowych i płytki sterownika i sprawdzeniu czy na złączu J2 płytki układów analogowych nie ma niebezpiecznych napięć można oba układy połączyć. Do połączenia obu modułów jest przeznaczona 10-przewodowa taśma przewodów z zaciskniętymi złączami IDC10.

Testowanie kompletnego zasilacza można rozpocząć od ustawiania napięcia wyjściowego. Sterownik mierzy napięcie wyjściowe i jeżeli dzielnik pomiarowy został dobrze wyregulowany, to wyświetlana na

wyświetlaczu wartość napięcia powinna być równa napięciu zmierzonemu przez multimetr na zaciskach wyjściowych złącza WY. Potem obciążamy wyjście, najlepiej rezystorem o dużej mocy i porównujemy wskazania pomiaru prądu na wyświetlaczu z prądem mierzonym przez multimetr. Przy okazji obserwujemy, czy po obciążeniu napięcie wyjściowe jest stabilne (nie zmienia się). Oczywiście pod warunkiem, że prąd wyjściowy nie jest ograniczany przez ograniczenie prądowe. W kolejnym kroku ustawiamy próg zabezpieczenia prądowego. Po obciążeniu wyjścia prąd wyjściowy nie powinien być wyższy niż ustawiony próg. Napięcie wyjściowe nie wzrasta przy próbie podwyższenia, jeżeli prąd wyjściowy osiągnie wartość równą progowi zabezpieczenia.

Jeżeli prąd wyjściowy przekroczy próg zabezpieczenia prądowego ustawionego w funkcji *Set Overcurrent*, to przekaźnik RE1 powinien odłączyć wyjście zasilacza od obciążenia. Ponowne załączenie wyjścia jest możliwe po zmniejszeniu obciążenia i przycisnięciu przycisku KAS.

Po ustawieniu progu napięciowego funkcją *Set overvoltage* sterownik nie pozwoli na ustawienie napięcia wyjściowego wyższego niż ten próg. W czasie normalnej pracy trzeba napięcie progowe ustawić na poziomie powyżej +24 V.

Na **fotografii 9** pokazane wewnątrz modelowego zasilacza. Umieściłem go w typowej łatwej do kupienia obudowie. Zamieściły się tu wszystkie elementy: transformator o mocy ok. 100 VA, transformator do zasilania układów pomocniczych, obie płytki, wyłącznik sieciowy i gniazdo bezpiecznika. Radiator z tranzystorem mocy został przykręcony na zewnątrz do ścianki tylnej obudowy.

Na koniec

Przedstawiony tu zasilacz został zaprojektowany do używania w typowych warunkach domowego warsztatu elektronika. Przy doborze elementów starałem się by jak najwięcej z nich było typowych, łatwych do kupienia i tanich. Mikrokontroler STM32F100RBT6B można bez problemu zakupić w ilościach detalicznych i w dobrej cenie u niejednego z krajowych dostawców. Najdroższym elementem jest transformator. Poprawne działanie zasilacza wymaga napięć przemiennych $2 \times 14 \dots 15$ V przy obciążeniu większym lub równym 3A i dodatkowego napięcia $30 \dots 33$ V o obciążalności 200 mA. Trudno znaleźć w domowych zasobach transformator o takich parametrach. Można zastosować 2 oddzielne transformatory i ja tak zrobiłem. Można też zamówić wykonanie takiego elementu w specjalistycznym zakładzie.

Ze względów bezpieczeństwa od strony sieci zasilającej 230VAC zasilacz musi być wyposażony w bezpiecznik topikowy. Jego wartość będzie zleżała od mocy transformatora – na próbę można użyć bezpiecznika szybkiego o prądzie znamionowym 2 A.

Umieszczenie zasilacza w metalowej obudowie wymaga odpowiedniej, podwójnej izolacji obwodów 230 VAC, stosowania 3-przewodowego kabla zasilającego z odpowiednią wtyczką, dołączenia przewodu neutralnego do obudowy zasilacza i zasilanie z gniazdka z prawidłowo połączonym kołkiem uziemiającym. Sieciowy przewód zasilający powinien być wprowadzony do obudowy przez specjalny gumowy lub plastikowy przepust. Nieprzestrzeganie tych zaleceń grozi porażeniem i jeżeli ktoś nie ma doświadczenia w tego typu konstrukcjach, to lepiej jest poprosić o pomoc kogoś bardziej doświadczonego. Dla obudowy z materiału izolacyjnego wymagania są mniejsze, ale tu również należy zadbać o prawidłowe wykonanie obwodów 230 VAC. Trzeba też wspomnieć, że takie elementy jak: transformator sieciowy, wyłącznik sieciowy i oprawa bezpiecznikowa, muszą być dobrej jakości i przystosowane do przy napięciu co najmniej 250 VAC.

Tomasz Jabłoński, EP