



**Podstawowe parametry:**

- zakres pomiarowy mierzonego napięcia: 0...34 V,
- rozdzielczość pomiaru napięcia: 10 mV,
- zakres pomiarowy mierzonego prądu: 0...10 A,
- rozdzielczość pomiaru prądu: 1 mA,
- zakres pomiarowy mierzonej mocy: 0...340 W,
- rozdzielczość pomiaru mocy: 100 mW,
- zakres pomiarowy dostarczonego ładunku: 0...100 Ah,
- rozdzielczość pomiaru ładunku: 10 mAh,
- częstotliwość rejestracji danych na karcie SD: 5 razy na sekundę,
- maksymalny pobór prądu ze źródła napięcia zasilającego: 40 mA.

\* **Uwaga!** Elektroniczne zestawy do samodzielnego montażu. Wymagana umiejętność lutowania! Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wylutować w dołączonej płytce drukowanej (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wylutowane w płytce PCB),
  - wersja [A] – płytka drukowana bez elementów i dokumentacji.
- Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- wersja [A+] – płytka drukowana [A] + zaprogramowany układ [UK] i dokumentacja,
  - wersja [UK] – zaprogramowany układ.

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

- AVT5969 powerMonitor – miniaturowy miernik parametrów elektrycznych (EP 2/2023)
- AVT5954 Warsztatowy Multi Tester (EP 10/2022)
- AVT5766 Ampera – miniaturowy miernik prądu stałego (EP 5/2020)
- AVT5672 USBtester – monitor parametrów zasilania USB (EP 5/2019)
- Projekt 232 Miliamperomierz TRMS (EP 9/2017)
- AVT5507 Miernik UIPTR (EP 7/2015)
- AVT5488 Kontroler obciążenia portu USB (EP 2/2015)
- AVT1823 Monitor prądu USB (EP 8/2014)
- AVT5399 Dwukanałowy multimetr panelowy (EP 6/2013)
- AVT5386 Podwójny woltomierz i amperomierz (EP 3/2013)

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik PDF! Podczas składania zamówienia upewnij się, którą wersję zamawiasz!  
<http://sklep.avt.pl>

W przypadku braku dostępności na stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: [kity@avt.pl](mailto:kity@avt.pl).

W ofercie AVT\*

**AVT6027**

# powerTracker

## Rejestrator parametrów elektrycznych

*Pomiar parametrów zasilania urządzenia elektronicznego wymaga zastosowania dwóch osobnych mierników (woltomierza i amperomierza) lub laboratoryjnego miernika mocy. Pierwsze rozwiązanie wiąże się z koniecznością żmudnego podłączania przewodów w odpowiedniej konfiguracji, drugie jest przeważnie zarezerwowane tylko dla profesjonalnych laboratoriów pomiarowych. Opisane urządzenie umożliwia rejestrację napięcia i prądu stałego, mocy oraz ładunku dostarczonego do testowanego odbiornika – doskonale sprawdzi się zatem w rozmaitych aplikacjach warsztatowych oraz serwisowych.*



Pomysł na ten projekt podsunął mi jeden z naszych Czytelników (przy okazji: pozdrowienia dla Pana Janusza), który w mailu skierowanym do mnie wyrażał uznanie dla mojej inwencji twórczej – jednym z jej przejawów miał być projekt opublikowany w numerze 1/2023 „Elektroniki Praktycznej”, urządzenie o nazwie powerMonitor. Zadaniem tej konstrukcji było monitorowanie parametrów elektrycznych odbiornika energii prądu stałego. Niestety, a może i na szczęście, do zachwyków dodana została przysłowiowa łyżka dziegciu. Wspomniany Czytelnik zauważył, iż tego typu monitor powinien być wyposażony w opcję rejestrowania mierzonych wartości elektrycznych, by można było je później poddać analizie – gdyż ich prezentacja na tak małym wyświetlaczu jest dyskusyjna. Cóż, nie sposób nie zgodzić się z tego rodzaju tezą, w związku z czym

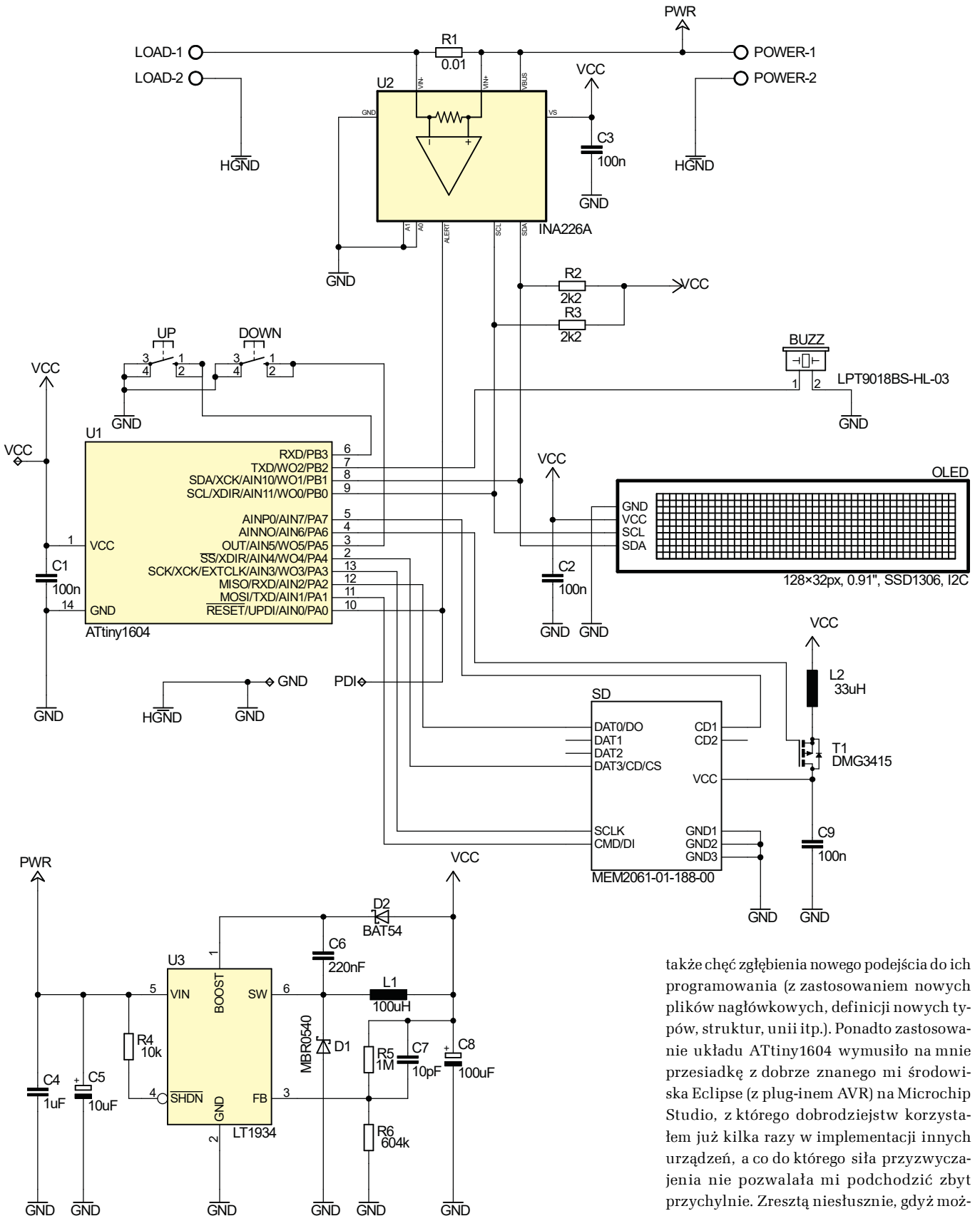
zabrałem się ochoczo za prace projektowe. Postanowiłem zbudować urządzenie o zbliżonej funkcjonalności, lecz wyposażone w możliwość rejestracji danych na karcie micro SD. W ten prozaiczny sposób narodził się projekt o nazwie powerTracker, który jest tematem niniejszego artykułu. Przystępując do prac konstrukcyjnych, poczyniłem wstępne założenia co do funkcjonalności urządzenia, które prezentują się następująco:

- możliwość pomiaru napięcia na zaciskach odbiornika,
- możliwość pomiaru prądu i mocy pobieranej przez odbiornik,
- możliwość pomiaru ładunku przekazanego do odbiornika,
- możliwość generowania alarmów po przekroczeniu konfigurowalnych parametrów – niezależnie dla każdej z wielkości elektrycznych (alarm powyżej lub poniżej ustawionej wartości),
- możliwość rejestracji parametrów elektrycznych na karcie micro SD sformatowanej zgodnie z systemem plików FAT16/FAT32.

Jako że, konstruuąc poprzednie urządzenie, zdobyłem niezbędne doświadczenie

w kwestii wykorzystywanych peryferiów, postanowiłem użyć ich i tym razem. Jedynym dodatkowym problemem, z jakim musiałem się zmierzyć, była implementacja obsługi karty standardu SD oraz systemu plików, co z góry wykluczało zastosowanie prostego mikrokontrolera ATtiny84 z poprzedniego projektu. Tym razem musiałem sięgnąć po bardziej nowoczesny procesor, którego zasoby sprzętowe sprostająby stawianemu przed nim zadaniu. Tak oto powstał system mikroprocesorowy, który pokazano na **rysunku 1**.

Jak widać, mamy tu bardzo prosty układ, którego sercem jest niewielki, acz nowoczesny mikrokontroler firmy Microchip (dawniej Atmel) typu ATtiny1604 (należący do nowej rodziny 0-series AVR), taktowany wewnętrznym generatorem RC o częstotliwości 3,333 MHz. Odpowiada on za sprzętową implementację interfejsu I<sup>2</sup>C (nazywanego przez firmę Microchip mianem TWI), przy użyciu którego mikrokontroler realizuje obsługę układu INA226 – będącego specjalizowanym, bardzo dokładnym, 16-bitowym, różnicowym przetwornikiem ADC – oraz niewielkiego, ale bardzo efektywnego wyświetlacza OLED o rozdzielczości 128×32 px,



Rysunek 1. Schemat ideowy urządzenia powerTracker

stanowiącego element graficznego interfejsu użytkownika. Wybór tego konkretnego mikrokontrolera z szerokiej rodziny układów firmy Microchip podyktowany był chęcią utrzymania małego footprintu obudowy (podobnie jak ATtiny84) oraz wymogiem

dostępności większej ilości pamięci Flash i RAM. Inną, trochę mniej oczywistą przyczyną wyboru tego układu była chęć poznania nowej rodziny mikrokontrolerów AVR oraz sposobu obsługi zaktualizowanych peryferiów, w jakie je wyposażono – ale

także chęć zgłębienia nowego podejścia do ich programowania (z zastosowaniem nowych plików nagłówkowych, definicji nowych typów, struktur, unii itp.). Ponadto zastosowanie układu ATtiny1604 wymusiło na mnie przesiadkę z dobrze znanego mi środowiska Eclipse (z plug-inem AVR) na Microchip Studio, z którego dobrodziejstw korzystałem już kilka razy w implementacji innych urządzeń, a co do którego siła przyzwyczajenia nie pozwalała mi podchodzić zbyt przychylnie. Zresztą niesłusznie, gdyż możliwość debugowania czy symulowania pracy mikrokontrolera jest nie do przecenienia.

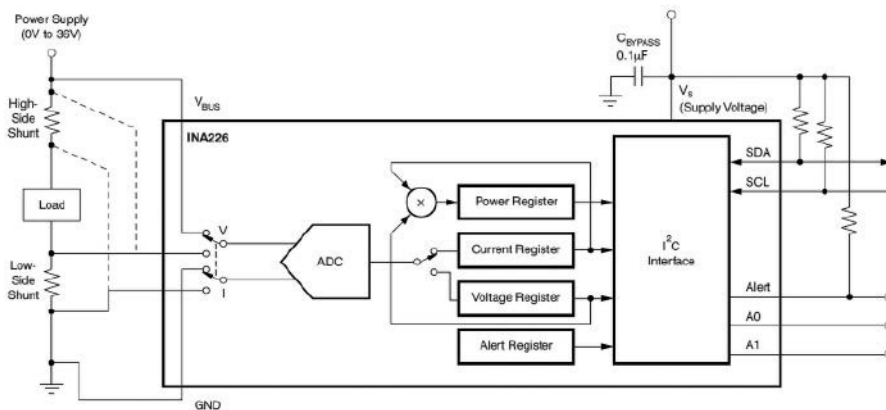
Mikrokontroler w naszym systemie odpowiedzialny jest dodatkowo za obsługę 2 mikroswitchy UP i DOWN. W celu eliminacji drgań styków używa on wbudowanego 16-bitowego układu czasowo-licznikowego TCBO, pracującego w trybie Periodic Interrupt. Obsługuje również buzzer piezoelektryczny

(bez wewnętrznego generatora), co realizowane jest z kolei z użyciem 16-bitowego układu czasowo-licznikowego TCA0, pracującego w trybie Frequency Generation (i generującego przebieg prostokątny o częstotliwości 4 kHz na wyprowadzeniu W02 (PB2), zasilający wspomniany buzzer). Wspomniany wcześniej, specjalizowany przetwornik ADC mierzy spadek napięcia na rezystorze szeregowym R1 (10 mΩ), dzięki czemu możliwe jest wyznaczenie prądu pobieranego przez badane urządzenie. Nie jest to jednak zwykły, zewnętrzny przetwornik ADC, jakich wiele na rynku, a specjalizowany układ przeznaczony do pomiaru prądu, napięcia i mocy urządzeń zasilanych napięciem stałym. Jako że jest to element dość wyjątkowy, warto choćby skrótowo zaznajomić się z jego specyfikacją. Tak jak napisano wcześniej, układ INA226 produkcji firmy Texas Instruments jest bardzo dokładnym, 16-bitowym, różnicowym przetwornikiem pomiarowym ADC, przeznaczonym do współpracy z bocznikiem rezystancyjnym. Układ ten wyróżnia się następującymi cechami użytkowymi:

- szeroki zakres napięć zasilania 2,7...5,5 V,
- bardzo duża dokładność pomiaru rzędu 0,1%,
- możliwość pracy w systemach o szerokim zakresie napięć szyny zasilającej 0...36 V,
- możliwość pracy w konfiguracjach Low-side i High-side,
- bezpośredni pomiar napięcia, prądu i mocy,
- konfigurowalny czas przetwarzania wbudowanego przetwornika ADC,
- konfigurowalna funkcja uśredniania pomiarów,
- dwa tryby pracy wbudowanego przetwornika ADC: ciągły i wyzwalany na żądanie,
- możliwość alarmowania po przekroczeniu danego poziomu prądu, napięcia szyny zasilającej odbiornik bądź mocy pobieranej przez odbiornik.

Jak widać, układ INA226 idealnie wpisuje się w wymagania naszej aplikacji, oferując niespotykaną dotąd funkcjonalność i dokładność pomiarów. Schemat blokowy tego peryferium pokazano na **rysunku 2**.

Opisywany układ dokonuje ciągłego (lub wyzwalanego manualnie przez aplikację użytkownika) pomiaru dwóch wartości napięć:



Rysunek 2. Schemat blokowy układu INA226

napięcia szyny zasilającej odbiornik (**VBUS**) oraz napięcia na zaciskach bocznika rezystancyjnego (**VSHUNT**), włączonego w szereg z odbiornikiem. Na podstawie tych dwóch wielkości oraz zawartości rejestru konfiguracyjnego CALIBRATION (którego zawartość zależy od wymaganej rozdzielczości pomiaru i parametrów zastosowanego bocznika rezystancyjnego), układ oblicza prąd oraz moc pobieraną przez odbiornik, po czym udostępnia je aplikacji użytkownika, ładując wyniki obliczeń do stosownych rejestrów konfiguracyjnych, jak również ustawiając dedykowane flagi zakończenia konwersji. Ponadto, dzięki wyposażeniu w grupę specjalnych rejestrów konfiguracyjnych (odpowiedzialnych za porównywanie zmierzonych i obliczonych wartości z wartościami progowymi) oraz dedykowane wyprowadzenie oznaczone jako **ALERT**, umożliwia generowanie alarmów po przekroczeniu zdefiniowanych przez użytkownika progów dotyczących: napięcia szyny zasilającej, napięcia na boczniku pomiarowym i mocy pobieranej przez odbiornik. Co więcej: producent tego peryferium wyposażył je w możliwość niezależnej konfiguracji czasu przetwarzania przetwornika ADC, oddzielnie dla napięcia szyny zasilającej i napięcia bocznika rezystancyjnego, a także możliwość uśredniania tychże wielkości spośród wielu, kolejno następujących pomiarów. Dzięki takiemu podejściu zwiększono wydatnie funkcjonalność układu i możliwość dostosowania trybu jego pracy do wymagań konkretnej aplikacji. Pamiętać należy jedynie, że wydłużenie czasu przetwarzania wbudowanego przetwornika ADC wydatnie podwyższa uzyskaną dokładność

miaru, zaś uśrednianie większej liczby próbek zdecydowanie zwiększa odstęp sygnału od szumu, w związku z czym w rzeczywistych aplikacjach należy dobrać maksymalne i możliwe do zaakceptowania wartości tychże parametrów, kierując się dla przykładu szybkością zmian badanych przebiegów jako kryterium wyjściowym. Co oczywiste, by poznać wszystkie możliwości drżące w układzie INA226, należałoby sięgnąć do jego noty aplikacyjnej lub do... mojego artykułu wspomnianego wcześniej. Szczegółowo omawiam w nim wszelkie zagadnienia implementacyjne, dlatego nie będę ich powielał w niniejszym opracowaniu, tym bardziej że jest to dość obszerna lektura.

Uważny Czytelnik zauważy z pewnością dość rozbudowany blok zasilający, zbudowany z użyciem nowoczesnej, scalonej przetwornicy step-down typu LT1934 firmy Analog Devices, która w zastosowanej wersji może dostarczać prąd o wartości 300 mA. Dlaczego zastosowałem przetwornicę i w dodatku tak nietypową? To proste. Chciałem, by nasze urządzenie było zasilane z tego samego źródła, co badany odbiornik, czyli de facto było włączane w szereg pomiędzy źródłem zasilania a badanym odbiornikiem. Jako że założyłem sobie dość szeroki zakres napięć dla źródeł zasilania, niemożliwe stało się zastosowanie zwykłego stabilizatora liniowego (nawet LDO) w celu zasilania systemu mikroprocesorowego z uwagi na zbyt dużą moc traconą i problem z odprowadzeniem ciepła (nie mówiąc już nawet o sprawności takiego układu). Okazało się dość szybko, iż wśród dostępnych na rynku półprzewodników trudno jest znaleźć taki, który spełniałby

**Wykaz elementów:**

**Rezystory:**

- R1: pomiarowy 10 mΩ 1 W 1% (SMD 1206)
- R2, R3: 2,2 kΩ (SMD 0805)
- R4: 10 kΩ (SMD 0805)
- R5: 1 MΩ 1% (SMD 0805)
- R6: 604 kΩ 1% (SMD 0805)

**Kondensatory:**

- C1...C3, C9: 100 nF (SMD 0805, X7R)
- C4: 1 µF (SMD 0805, X7R)
- C5: 10 µF/50 V typu 50SVF10M PANASONIC (polimerowy, LOW ESR, SANYO SMD C6)

- C6: 220 nF (SMD 0805, X7R)
- C7: 10 pF (SMD 0805, X7R)
- C8: 100 µF/10 V (tantalowy, LOW ESR, C/6032-28R)

**Półprzewodniki:**

- U1: ATtiny1604 (SOIC14)
- U2: INA226A (MSOP-10)
- U3: LT1934 (SOT23)
- T1: DMG3415 (SOT23)
- D1: MBR0540 (SOD123)
- D2: BAT54 (SOT23)
- OLED: wyświetlacz OLED 128x32 px, 0,91", sterownik

SSD1306, magistrala I2C, wymiary 38 x 12 mm

**Pozostałe:**

- L1: 100 µH typu DLG-0504-101
- L2: 33 µH (SMD 0805)
- UP, DOWN: microswitch SMD typu DTSM-65N-V-B lub podobny
- BUZZ: przetwornik piezoelektryczny SMD typu LPT9018BS-HL-03-4.0-12-R
- LOAD, POWER: złącze śrubowe AK500/2
- SD: gniazdo karty micro SD typu MEM2061-01-188-00-A (GLOBAL CONNECTOR TECHNOLOGY)

oczekiwane założenia (napięcie wejściowe rzędu 34 V i wyjściowe 3,3 V), a do tego występował w obudowie, która będzie łatwa do montażu przez amatora. Prawdę mówiąc, była to jedyna scalona przetwornica, która spełniała wszystkie wstępne założenia projektowe. Nie będę w tym miejscu wchodził w szczegóły implementacyjne, gdyż stosowne informacje znajdziemy w drobiazgowej dokumentacji producenta układu – zwłaszcza że zastosowane rozwiązanie jest typową implementacją zaproponowaną w nocie aplikacyjnej, zapewniającą szeroki zakres napięć zasilających (3,3...34 V) oraz maksymalny, dopuszczalny prąd obciążenia rzędu 300 mA, co z zapasem zaspokaja zapotrzebowanie energetyczne urządzenia (zresztą największy prąd pobiera wyłącznie karta SD w trybie zapisu). Niemniej jednak warto wspomnieć o pewnym ograniczeniu tej przetwornicy. Otóż układ LT1934 może pracować poprawnie już przy napięciu wejściowym rzędu 3,3 V, ale do startu samej przetwornicy niezbędne jest nieco większe napięcie – mianowicie 4,5 V. Gdy przetwornica uruchomi się, napięcie zasilające można zmniejszyć do wspomnianego poziomu, jednak to ograniczenie należy mieć zawsze na uwadze. Co prawda można mu zaradzić, modyfikując minimalnie zaproponowaną implementację poprzez podłączenie anody diody D2 do napięcia wejściowego (VIN) zamiast wyjściowego (VCC). W takim przypadku ograniczymy jednak zakres napięć zasilających: tym razem od góry do maksymalnej wartości 20 V, co wynika z dopuszczalnego napięcia na wyprowadzeniu BOOST przetwornicy, które w takim układzie wynosi w przybliżeniu  $2 \times V_{IN}$  i nie może przekraczać wartości 40 V. Innym rozwiązaniem jest pozostanie przy schemacie pierwotnym i wstępne obciążenie wyjścia przetwornicy prądem minimalnym ok. 12 mA, co zapewni poprawny start układu nawet przy napięciu wejściowym 3,3 V, nie ograniczając tym samym maksymalnego poziomu napięcia wejściowego, lecz znacznie pogarszając sprawność układu (połowa prądu zostanie w tym przypadku „zmarowana” na podgrzewanie rezystora obciążającego). Ostateczne rozwiązanie pozostawiam Czytelnikom, gdyż

każdy musi określić, na jakie ograniczenia jest skłonny sobie pozwolić. Ja zdecydowałem się na wersję z implementacją producenta układu. Z myślą o dociekliwych Czytelnikach dodam, iż wartość napięcia wyjściowego ustalono na 3,3 V. Co ważne, już w tym miejscu chciałbym zaznaczyć, że w celu zapewnienia optymalnej pracy przetwornicy (w tym maksymalizacji jej sprawności) niezbędne jest zastosowanie odpowiedniej jakości kondensatorów (zwłaszcza elektrolitycznych o niskim ESR), co znajduje odzwierciedlenie w opisie listy elementów. Nie warto bagatelizować tego zagadnienia.

Przejdźmy zatem do kwestii implementacyjnych, które w tym przypadku ograniczę do opisu problemów związanych z obsługą karty SD oraz systemu plików FAT16/FAT32, gdyż pozostałe zagadnienia programistyczne zostały już poruszone w artykule dotyczącym poprzedniego mojego urządzenia. Jak zapewne się domyślicie, obsługa kart SD (w naszym przypadku micro SD) nie jest zagadnieniem łatwym, a jeśli dodać do tego jeszcze sam system plików FAT16/FAT32, zaczyna robić się nieciekawie. Na szczęście ktoś zrobił to wcześniej za nas i to na tyle dobrze (a przy okazji uniwersalnie), że nawet firma Microchip w swoich przykładach podaje wykorzystanie tego właśnie opracowania (nota aplikacyjna AN2799). Mowa o dobrze znanym module Petit FatFs autorstwa elm-chan, który jest uproszczoną wersją modułu FatFS tego samego autora, przeznaczoną na mikroprocesory o niewielkich zasobach pamięci Flash i RAM. Wersja ta ma co prawda pewne ograniczenia, ale za to można ją bez problemu uruchomić na tak skromnych mikrokontrolerach, jak rodzina Tiny AVR, gdyż wymaga jedynie 44 bajtów dostępnej pamięci RAM! Niesamowite, nieprawdaż? Ograniczenia, o których mowa, to:

- brak możliwości tworzenia nowych plików (obsługiwane są wyłącznie pliki istniejące),
- brak możliwości zmiany rozmiaru plików,
- data/godzina utworzenia/modyfikacji pliku nie ulega zmianie nawet po operacji zapisu,

- operacja zapisu musi zaczynać się i kończyć w ramach tego samego sektora,
- atrybut „tylko do odczytu” pliku nie blokuje operacji zapisu.

Jak widać, są to ograniczenia o niezbyt wielkim ciężarze gatunkowym, lecz jeśli w Waszym rozwiązaniu wydają się istotne, zawsze możecie sięgnąć po moduł FatFS (tego samego autora), który – co oczywiste – wymaga już znacznie większych zasobów sprzętowych. Dla naszego projektu ograniczenia takie, jak wyżej opisane, nie mają żadnego praktycznego znaczenia, a skutkiem ich istnienia jest konieczność przygotowania na karcie SD „pustego” pliku o predefiniowanej nazwie i odpowiedniej wielkości (by można było w nim zapisywać dane – wszak moduł Petit FatFs nie może zmienić jego rozmiaru). Co więcej, firma Microchip przygotowała przykład wykorzystania tego modułu – możemy go wygenerować ze środowiska Microchip Studio (przykład AVR4276), a następnie dołączyć do nowego projektu. Co prawda wspomniane oprogramowanie przygotowano dla mikrokontrolera ATtiny817, lecz bez większego trudu powinniśmy dostosować je do używanego typu układu. Tak naprawdę modyfikacji wymaga głównie plik `diskio_avr.c` (i związany z nim `diskio.h`), zawierający niskopoziomowe funkcje dostępu do medium komunikacyjnego (SPI) oraz plik `pfconf.h`, w którym włączyć musimy kompilację warunkową niezbędnych funkcji bibliotecznych oraz obsługę używanych systemów plików (za pomocą stosownych definicji), co ma oczywisty wpływ na rozmiar skompilowanego projektu. Jak widzicie, zagadnienie nie wydaje się szczególnie skomplikowane, a mimo to na wielu forach dyskusyjnych znaleźć można setki wątków, w ramach których użytkownicy opisują swoje nieudane próby uruchomienia wspomnianego oprogramowania. Z tego właśnie powodu zilustruję niezbędne modyfikacje przykładu firmy Microchip, dostosowane do naszego projektu (i mikrokontrolera), rozszerzające jego funkcjonalność o możliwość detekcji obecności nośnika w gnieździe karty SD oraz sterowania zasilaniem

```
//Definicje portów interfejsu SPI karty SD -> DO, DI, SCK, CS
#define SD_PORT_NAME PORTA
#define SD_SCK_MASK PIN3_bm //PA3 -> SCK karty SD
#define SD_DO_MASK PIN2_bm //PA2 -> MISO -> DO karty SD
#define SD_DI_MASK PIN1_bm //PA1 -> MISO -> DI karty SD
#define SD_CS_MASK PIN4_bm //PA4 -> CS karty SD
#define SD_DO_PINCTRL_REG PIN2CTRL

//Definicje pozostałych portów karty SD -> DETECT, POWER
#define SD_POWER_MASK PIN6_bm //PA6 -> POWER karty SD (0 -> zasilanie włączone, 1 -> zasilanie wyłączone)
#define SD_DETECT_MASK PIN7_bm //PA7 -> CARD DETECT karty SD (Karta obecna wgnieździe -> 1)
#define SD_DETECT_PINCTRL_REG PIN7CTRL
#define SD_POWER_ON SD_PORT_NAME.OUTCLR = SD_POWER_MASK
#define SD_POWER_OFF SD_PORT_NAME.OUTSET = SD_POWER_MASK
#define SD_IS_PRESENT (SD_PORT_NAME.IN & SD_DETECT_MASK)

//Definicje dla portu CS
#define SELECT() SD_PORT_NAME.OUTCLR = SD_CS_MASK //CS = 0
#define DESELECT() SD_PORT_NAME.OUTSET = SD_CS_MASK //CS = 1
#define SELECTING ((SD_PORT_NAME.DIR & SD_CS_MASK) && !(SD_PORT_NAME.OUT & SD_CS_MASK))
```

Listing 1. Zmieniony względem oryginału fragment zawartości pliku nagłówkowego `diskio.h` modułu obsługującego karty SD (w ramach modułu Petit FatFs)



```
static void init_spi(void)
{
    //Porty MOSI, CS, SCK i POWER jako wyjściowe, zaś MISO i DETECT, jako wejściowe podciągnięte do VCC
    SD_PORT_NAME_DIRSET = SD_DI_MASK|SD_SCK_MASK|SD_CS_MASK|SD_POWER_MASK;
    SD_PORT_NAME_SD_DO_PINCTRL_REG = PORT_PULLUPEN_bm;
    SD_PORT_NAME_SD_DETECT_PINCTRL_REG = PORT_PULLUPEN_bm;

    SPI0.CTRLA = SPI_MASTER_bm|SPI_PRESC_DIV16_gc; //SPI Master, fclk = 208 kHz dla F_CPU = 3.333 MHz
    SPI0.CTRLB = SPI_SSD_bm; //Pin SS kontrolowany przez oprogramowanie
    SPI0.CTRLA |= SPI_ENABLE_bm; //Włączenie modułu SPI

    //Włączamy zasilanie karty SD i oczekujemy chwilę na start sterownika karty
    SD_POWER_ON;
    _delay_ms(100);
}
```

**Listing 2.** Zmieniony względem oryginału fragment zawartości pliku `diskio_avr.c`, pokazujący zredefiniowane ciało funkcji inicjalizującej interfejs SPI (w ramach modułu Petit FatFs)

wspomnianego medium. Sterowanie zasilaniem karty znajduje zastosowanie w przypadku problemów z poprawną jej inicjalizacją, co zdarzyć się może z różnych powodów. W takim wypadku możemy wyłączyć, a później włączyć jej zasilanie, następnie od czekać czas niezbędny na inicjalizację jej wewnętrznego sterownika, po czym ponownie spróbować zainicjować go zgodnie z naszymi oczekiwaniami. Na **listingu 1** pokazano zmieniony względem oryginału fragment zawartości pliku nagłówkowego `diskio.h`, w ramach którego zebrano definicje wypro wadzeń karty SD oraz nazwy stosownych rejestrów sterujących.

Dalej, na **listingu 2**, zaprezentowano zmieniony względem oryginału fragment zawartości pliku `diskio_avr.c`, w ramach którego zredefiniowano ciało funkcji inicjalizującej interfejs SPI.

I na koniec, w pliku `pfconf.h`, włączamy niezbędne opcje kompilacji, co pokazano na **listingu 3**.

Prawda, że proste? Na koniec, już w ramach aplikacji urządzenia `powerTracker`, zaimplementowano funkcję inicjalizującą kartę SD wraz z niezbędnym „montażem” woluminu oraz obsługą systemu plików, której ciało pokazano na **listingu 4**.

To wszystko, jeśli chodzi o niezbędne modyfikacje modułu Petit FatFs. Na koniec zagadnień implementacyjnych opiszę jeszcze prosty moduł do obsługi interfejsu I<sup>2</sup>C, wykorzystujący sprzęg TWI znajdujący się na pokładzie mikrokontrolera ATtiny1604. Przyznam, że nie jest to szczególnie nowatorskie rozwiązanie, jednak interfejs TWI w nowych mikrokontrolerach AVR z serii 0 różni się znacząco (pod względem sposobu obsługi i rozmieszczenia oraz znaczenia rejestrów konfiguracyjnych) od starszych wersji tychże procesorów, przez co uważam, że opracowanie, takie jak powyższe, jest niezbędne. Warto podkreślić, iż nie będzie to jednak rozwiązanie na wskroś uniwersalne, gdyż zależało mi na prostocie rozwiązań programistycznych – lecz w zastosowaniach tak łatwych, jak opisane tutaj, moim zdaniem okazuje się wystarczająco dobre. Podobnie jak poprzednio, stosowny driver (w tym używający przerwań TWI) możemy wygenerować automatycznie z poziomu środowiska Microchip Studio. Jak możecie jednak sami się

```
#define _USE_READ 0 /* Enable pf_read() function */
#define _USE_DIR 0 /* Enable pf_opendir() and pf_readdir() function */
#define _USE_LSEEK 1 /* Enable pf_lseek() function */
#define _USE_WRITE 1 /* Enable pf_write() function */

#define _FS_FAT12 0 /* Enable FAT12 */
#define _FS_FAT16 1 /* Enable FAT16 */
#define _FS_FAT32 1 /* Enable FAT32 */
```

**Listing 3.** Fragment zawartości pliku `pfconf.h` – kod pozwalający na włączenie stosownych opcji kompilacji (w ramach modułu Petit FatFs)

```
FRESULT initSdCard(void)
{
    DSTATUS Status;
    FRESULT Result;
    uint8_t Attempts = 1;

    //Najpierw sprawdzamy, czy karta SD jest w ogóle włożona do gniazda
    if(SD_IS_PRESENT)
    {
        //Inicjujemy naped fizyczny. Jeśli inicjalizacja się nie powiodła, to wyłączamy
        //i włączamy zasilanie karty, przy czym proces ponawiamy max. trzykrotnie
        do
        {
            Status = disk_initialize(); //Status powinien być RES_OK

            if(Status != RES_OK)
            {
                SD_POWER_OFF;
                _delay_ms(10);
                SD_POWER_ON;
                _delay_ms(100);
            }
            else
            {
                //Poprawnie zainicjowano kartę SD, więc zwiększamy prędkość magistrali SPI
                //-> fclk = 1.66 MHz dla F_CPU = 3.333 MHz
                SPI0.CTRLA = SPI_MASTER_bm|SPI_CLK2X_bm|SPI_PRESC_DIV4_gc|SPI_ENABLE_bm;
            }
        } while((Status != RES_OK) && (Attempts++ < 4));

        if(Status != RES_OK) return FR_DISK_ERR;

        //Montujemy wolumin
        Result = pf_mount(&fileSystem); //Result powinien być FR_OK, jeśli nie jest
        //taki to zwracamy numer błędu
        if(Result != FR_OK) return Result;

        //Otwieramy plik
        Result = pf_open("Tracker.txt"); //Result powinien być FR_OK, jeśli nie jest
        //taki to zwracamy numer błędu
        if(Result != FR_OK) return Result; else pf_lseek(fileOffset);

        return FR_OK;
    }
    else return FR_NOT_READY; //Karta nie jest włożona do gniazda
}
```

**Listing 4.** Funkcja inicjalizująca kartę SD wraz z niezbędnym „montażem” woluminu i obsługą systemu plików

przekonać, rozwiązanie proponowane przez producenta jest bardzo skomplikowane – jeśli bowiem ma pozostać w pełni uniwersalne, musi operować na dość wysokim poziomie abstrakcji. Zaczniemy od pliku nagłówkowego, którego ciało pokazano na **listingu 5**.

Dalej, na **listingu 6** zaprezentowano ciało funkcji inicjalizującej sprzęg TWI mikrokontrolera AVR 0-series.

Na **listingu 7** pokazano z kolei ciało funkcji odpowiedzialnej za wygenerowanie sygnału START interfejsu I<sup>2</sup>C i przesłanie adresu klienta (wraz z bitem kierunku R/W), zaś na **listingu 8** – ciało funkcji odpowiedzialnej

za wygenerowanie sygnału STOP interfejsu I<sup>2</sup>C.

I na koniec dwie kluczowe funkcje, pozwalające na zapis i odczyt bajtu poprzez interfejs I<sup>2</sup>C, których ciała pokazano odpowiednio na **listingach 9 i 10**.

Tyle – w telegraficznym skrócie – w kwestiach implementacyjnych. Przejdźmy zatem do schematu montażowego naszego urządzenia, zaprezentowanego na **rysunku 3**.

Jak widać, zaprojektowano bardzo zgrabną, dwustronną, niewielką płytkę drukowaną o nietypowym wyglądzie. Swoim kształtem dopasowana jest ona do obudowy opisanej

```
//Definicje portów magistrali I2C
#define I2C_PORT_NAME PORTB
#define I2C_SDA_PINCTRL_REG PIN1CTRL //PB1
#define I2C_SCL_PINCTRL_REG PIN0CTRL //PB0

//Definicje częstotliwości magistrali oraz czasu narastania zboczy
#define I2C_FREQUENCY 400000UL //Częstotliwość magistrali [Hz]
#define I2C_RISE_TIME 100UL //Czas narastania zboczy zależny od impedancji magistrali [ns]
#define I2C_BAUD (uint8_t)(((((float) F_CPU/(float) I2C_FREQUENCY)) - 10 - ((float) F_CPU * I2C_RISE_TIME/1000000))/2)

//Definicje bitu potwierdzenia (ACK)
#define NACK 0
#define ACK 1

//Definicje statusów wykonania funkcji
#define OK 0
#define FAILED 255
```

Listing 5. Plik nagłówkowy modułu obsługi interfejsu I<sup>2</sup>C mikrokontrolerów AVR 0-series

w dalszej części artykułu. Warto również podkreślić, że dla zminimalizowania rozmiaru obwodu drukowanego, przewidziano tutaj montaż elementów po obu stronach laminatu. Lutowanie urządzenia rozpoczynamy od warstwy TOP, na której w pierwszej kolejności umieszczamy wszystkie półprzewodniki. Proces ten najłatwiej wykonać przy użyciu stacji lutowniczej na gorące

```
void i2cInit(void)
{
    //Podciągnięcie portów SDA i SCL pod VCC
    I2C_PORT_NAME.I2C_SDA_PINCTRL_REG = PORT_PULLUPEN_bm;
    I2C_PORT_NAME.I2C_SCL_PINCTRL_REG = PORT_PULLUPEN_bm;

    TWI0.MBAUD = I2C_BAUD; //Ustawienie częstotliwości magistrali
    TWI0.MCTRLA = TWI_ENABLE_bm; //Włączenie modułu I2C -> Tryb Master bez
    //przerwań (tzw. pooling)
    TWI0.MSTATUS = TWI_BUSSTATE_IDLE_gc; //Domyślny tryb magistrali (Idle)
}
```

Listing 6. Ciało funkcji inicjalizującej sprzęg TWI mikrokontrolera AVR 0-series

```
uint8_t i2cStart(uint8_t Address)
{
    //Wysyłamy sygnał START z adresem klienta. Adres zawiera w sobie bit R/W (bit 0) decydujący o kierunku transmisji
    TWI0.MADDR = Address;

    while (!(TWI0.MSTATUS & (TWI_WIF_bm | TWI_RIF_bm))); //Czekamy na zakończenie operacji

    //Jeśli doszło do błędu lub arbitrażu na magistrali i dostęp do niej został utracony, zwracamy FAILED i czekamy na zwolnienie magistrali
    if(TWI0.MSTATUS & TWI_ARBLOST_bm)
    {
        while(!(TWI0.MSTATUS & TWI_BUSSTATE_IDLE_gc)); //Czekamy na zwolnienie magistrali
        return FAILED;
    }
    //Jeśli klient nie potwierdził swojego adresu, wysyłamy sygnał STOP i czekamy na zwolnienie magistrali
    else if(TWI0.MSTATUS & TWI_RXACK_bm)
    {
        TWI0.MCTRLB |= TWI_MCMD_STOP_gc; //Wysyłamy sygnał STOP
        while(!(TWI0.MSTATUS & TWI_BUSSTATE_IDLE_gc)); //Czekamy na zwolnienie magistrali
        return FAILED;
    }
    return OK;
}
```

Listing 7. Ciało funkcji odpowiedzialnej za wygenerowanie sygnału START interfejsu I<sup>2</sup>C i przesłanie adresu klienta

powietrze i odpowiednich stopów lutowniczych. Jeśli jednak nie dysponujemy tego rodzaju sprzętem, można również zastosować inną metodę. Najprostszym sposobem montażu elementów o tak dużym zagęszczeniu wyprowadzeń, niewymagającym jednocześnie posiadania specjalistycznego sprzętu, jest użycie zwykłej stacji lutowniczej, dobrej jakości cyny z odpowiednią ilością topnika oraz dość cienkiej plecionki lutowniczej, która umożliwi usunięcie nadmiaru cyny pomiędzy wyprowadzeń układów. Należy przy tym uważać, by nie uszkodzić termicznie tego rodzaju elementów. Następnie lutujemy elementy bierne, gniazdo karty micro

```
void i2cStop(void)
{
    TWI0.MCTRLB |= TWI_MCMD_STOP_gc; //Wysyłamy sygnał STOP
    while(!(TWI0.MSTATUS & TWI_BUSSTATE_IDLE_gc)); //Czekamy na zwolnienie magistrali
}
```

Listing 8. Ciało funkcji odpowiedzialnej za wygenerowanie sygnału STOP interfejsu I<sup>2</sup>C

```
uint8_t i2cWriteByte(uint8_t Byte)
{
    TWI0.MDATA = Byte;
    TWI0.MCTRLB = TWI_MCMD_RECVTRANS_gc; //Wysyłamy bajt danych
    while(!(TWI0.MSTATUS & TWI_WIF_bm)) //Czekamy na zakończenie transmisji

    //Jeśli doszło do błędu lub arbitrażu na magistrali i dostęp do niej został utracony
    //Zwracamy FAILED
    if(TWI0.MSTATUS & (TWI_ARBLOST_bm | TWI_BUSERR_bm)) return FAILED;

    return!(TWI0.MSTATUS & TWI_RXACK_bm); //Zwracamy bit ACK
}
```

Listing 9. Ciało funkcji odpowiedzialnej za zapis bajtu poprzez interfejs I<sup>2</sup>C

```
uint8_t i2cReadByte(uint8_t Ack)
{
    uint8_t Byte;

    while(!(TWI0.MSTATUS & TWI_RIF_bm)); //Czekamy na zakończenie odczytu
    Byte = TWI0.MDATA;
    //Wysyłamy sygnał ACK, gdy oczekujemy więcej danych (i wznawiamy transmisję) lub sygnał NACK
    if(Ack) TWI0.MCTRLB = TWI_MCMD_RECVTRANS_gc | TWI_ACKACT_ACK_gc; else TWI0.MCTRLB = TWI_ACKACT_NACK_gc;

    return Byte;
}
```

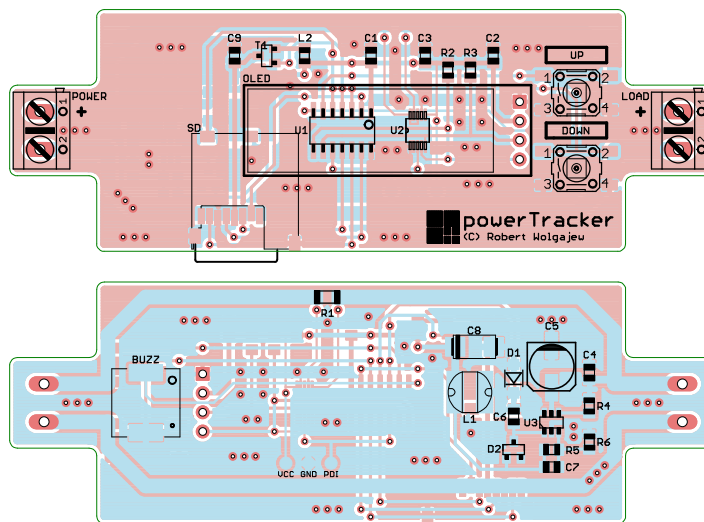
Listing 10. Ciało funkcji odpowiedzialnej za odczyt bajtu poprzez interfejs I<sup>2</sup>C

SD, przyciski UP i DOWN oraz złącza ARK o oznaczeniach POWER i LOAD, które docelowo będą wystawać poza obrys obudowy. Na samym końcu, do tak przygotowanej płytki, instalujemy wyświetlacz OLED, zwyczajnie wlotowując wyprowadzenia jego złącza typu goldpin w przewidziane do tego celu pola lutownicze (należy sprawdzić polaryzację zasilania), gdyż połączenia elektryczne zapewniają mu jednocześnie wystarczająco stabilny montaż mechaniczny. Wyświetlacz ten mocujemy w taki sposób, by górna płaszczyzna szkła wyświetlacza była umieszczona w odległości 5 mm od górnej powierzchni obwodu drukowanego (musimy przy tym zadbać, aby końcówki złącza goldpin nie wystawały ponad płaszczyznę szkła). Najłatwiej dokonać takiego montażu, przycinając wyprowadzenia złącza goldpin na wysokość 5 mm od płaszczyzny obwodu drukowanego. W tym momencie przechodzimy na warstwę BOTTOM, gdzie – jak poprzednio – w pierwszej kolejności przylutowujemy półprzewodniki, następnie elementy bierne, a na końcu buzzer piezoelektryczny SMD. Poprawnie zmontowany układ powinien działać tuż po włączeniu zasilania. Na **fotografii 1** pokazano zmontowane urządzenie powerTracker od strony warstwy TOP, tuż przed przylutowaniem wyświetlacza OLED, zaś na **fotografii 2** – to samo urządzenie od strony warstwy BOTTOM.

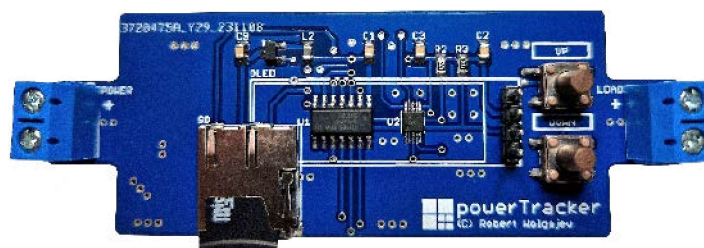
Warto podkreślić, że ścieżki przewodzące duże prądy (pomiędzy gniazdami POWER i LOAD) zaprojektowano w taki sposób, by możliwy był przepływ przez nie prądu stałego o wartości do 10 A. Co również istotne, zastosowany rodzaj bocznika pozwala na rozproszenie mocy rzędu 1 W, co spełnia założenia aplikacji. Jeżeli bocznik nagrzewałby się do nieakceptowalnej temperatury, należy zastosować element o większej dopuszczalnej mocy strat – bądź dwa elementy (o odpowiedniej rezystancji) połączone szeregowo lub równoległe. Przejdźmy teraz do zagadnień związanych z obsługą urządzenia.

### Obsługa

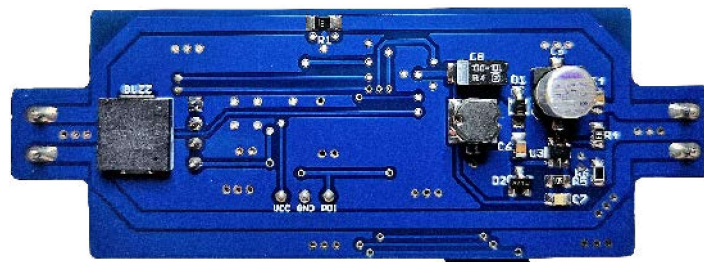
Projektując interfejs użytkownika modułu powerTracker, kierowałem się zarówno zasadą maksymalnego uproszczenia sposobu obsługi układu, jak i chęcią wyposażenia go w odpowiednią paletę możliwości. W realizacji tego celu posłużyłem się niewielkim, lecz bardzo atrakcyjnym wyświetlaczem OLED o rozdzielczości 128x32 px oraz dwoma mikroprzełącznikami oznaczonymi umownie jako UP i DOWN. Program obsługi aplikacji rozróżnia krótkie i długie naciśnięcie każdego z przycisków. Zgodnie z tymi założeniami powstał bardzo czytelny, graficzny interfejs użytkownika, w ramach którego wyświetlane są występujące 2 ekrany systemu menu o następujących funkcjonalnościach:



Rysunek 3. Schemat montażowy urządzenia powerTracker



Fotografia 1. Zmontowane urządzenie powerTracker, widok od strony warstwy TOP tuż przed przylutowaniem wyświetlacza OLED



Fotografia 2. Zmontowane urządzenie powerTracker, widok od strony warstwy BOTTOM

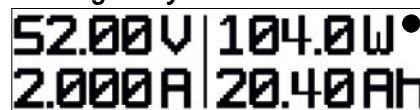
- ekran główny wskazujący wartość napięcia na zaciskach odbiornika, prądu pobieranego przez odbiornik, mocy i ładunku dostarczanego do odbiornika oraz ikonę prezentującą aktywność funkcji rejestracji pomiarów na karcie SD (migający symbol RECORD),
- ekran ustawień, w ramach którego dokonujemy ustawień funkcji alarmowania (w tym typu alarmu oraz wartości progowych).

Warto podkreślić, że w przypadku aktywnej funkcji rejestracji pomiarów na karcie SD próbki rejestrowane są 5 razy na sekundę – a po przekroczeniu wielkości pliku następuje wyzerowanie wskaźnika zapisu, przez co nowe pomiary nadpisują stare wartości. W związku z powyższym warto przygotować odpowiednio duży plik danych wypełniony, dla przykładu, samymi spacjami. Zapisywane wartości zgrupowane są w wiersze (zakreślane znakami CR i LF) oraz rozdzielone przecinkami tak, jak ma to miejsce w plikach typu CSV. Wspomniany plik, który należy

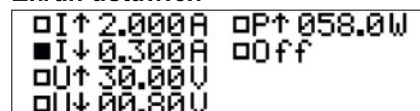
przygotować wcześniej, musi znajdować się w głównym katalogu karty i musi nosić nazwę „Tracker.txt”. Wygląd wszystkich ekranów systemu menu pokazano na **rysunku 4**.

Krótkie wciśnięcie przycisku UP powoduje wyzerowanie wskazania ładunku dostarczonego do odbiornika, zaś długie wciśnięcie tego przycisku – włączenie/wyłączenie procesu rejestracji danych na karcie SD, co sygnalizowane jest miganiem symbolu RECORD na ekranie głównym urządzenia. Włączeniu

#### Ekran główny



#### Ekran ustawień

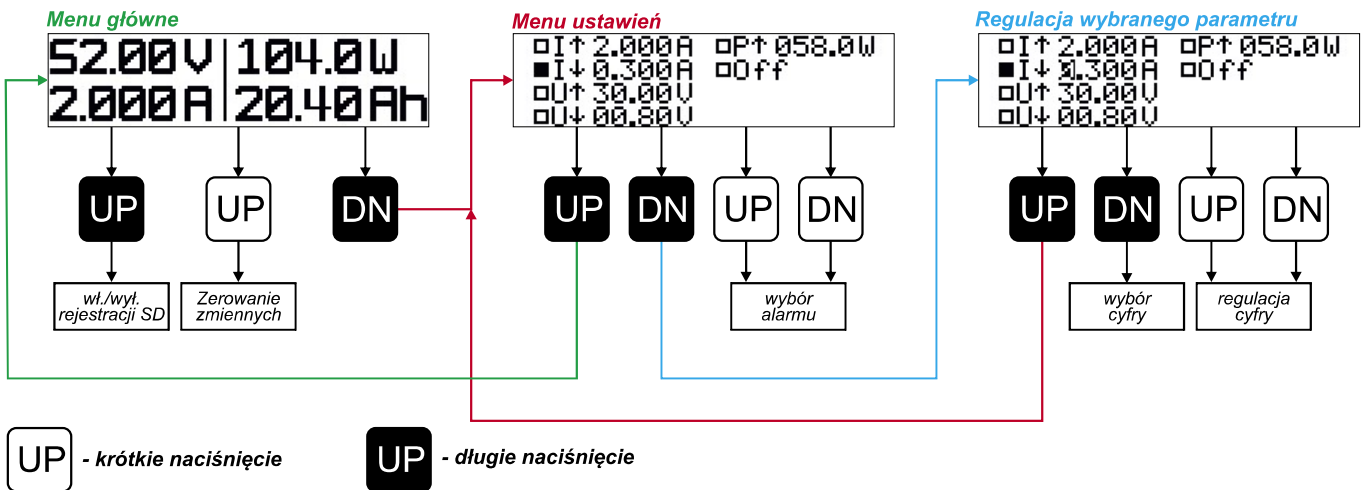


Rysunek 4. Wygląd wszystkich ekranów systemu menu urządzenia Tracker



**Tabela 1. Lista wszystkich możliwych błędów zgłaszanych przez urządzenie powerTracker wraz z opisem ich znaczenia dla systemu mikroprocesorowego**

Tekst błędu	Opis	Sugerowane działanie
SD disk error	Błąd sprzętowy obsługi karty SD lub nieprawidłowa struktura systemu plików	Sprawdź połączenia gniazda karty SD z systemem mikroprocesorowym. Sprawdź (w komputerze PC) strukturę systemu plików
SD not ready	Brak karty SD w gnieździe karty lub karta SD nie odpowiada na wysyłane komendy	Włóż kartę SD do gniazda lub zmień egzemplarz karty SD
No file	Brak pliku Tracker.txt w katalogu głównym karty SD	Utwórz plik Tracker.txt w katalogu głównym karty SD. Plik musi mieć odpowiednią do oczekiwanej pojemności wielkość pliku
No file system	Brak lub nieprawidłowy system plików na karcie SD	Sformatuj kartę SD zgodnie z systemem plików FAT16/FAT32

**Rysunek 5. Diagram prezentujący sposób obsługi urządzenia powerTracker**

procesu rejestracji danych na karcie SD towarzyszy inicjalizacja tejże karty i montaż systemu plików – ten ostatni może z różnych względów zakończyć się niepowodzeniem (np. jeśli karta micro SD nie znajduje się w złączu karty lub plik danych „Tracker.txt” nie istnieje). W takim wypadku zostanie wyświetlony stosowny komunikat błędu, zaś obsługa karty nie będzie możliwa. Listę wszystkich możliwych błędów zgłaszanych przez urządzenie powerTracker, wraz z opisem ich znaczenia dla systemu mikroprocesorowego, pokazano w tabeli 1.

Kontynuując zagadnienie obsługi systemu powerTracker, warto dodać, że długie przyciśnięcie przycisku DOWN na ekranie głównym powoduje z kolei przejście urządzenia do ekranu ustawień alarmowania. Powyższe zasady to wyłącznie drobny wycinek funkcjonalności

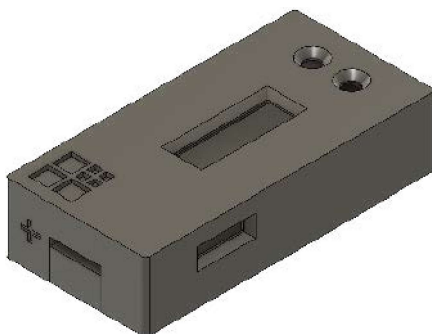
przycisków sterujących, gdyż ich funkcja w procesie obsługi urządzenia (wliczając w to rozróżnianie krótkiego bądź długiego przyciśnięcia) determinowana jest miejscem w systemie menu, w jakim znajduje się aktualnie urządzenie. Diagram prezentujący kompletny sposób obsługi urządzenia powerTracker pokazano na rysunku 5.

Na koniec warto podkreślić, że aktywna funkcja alarmu powoduje cykliczne sygnalizowanie (dźwiękiem wbudowanego buzzera) faktu przekroczenia ustawionego progu alarmowania (poniżej lub powyżej ustawionej wartości) do czasu ustąpienia warunków alarmu.

## Obudowa

I na sam koniec obiecany „ekstras” – mianowicie gotowy panel obudowy, opracowany w aplikacji do projektowania 3D przez mojego niezawodnego kolegę, **Bartłomieja Wawrzyszko**, zajmującego się hobbystycznie projektami tego rodzaju. Tak oto powstał model obudowy, którego widok 3D pokazano na rysunku 6.

Wspomniana obudowa tak naprawdę składa się z dwóch elementów: części górnej (pokazanej na rysunku 6), w której umieszczono otwory na elementy interfejsu użytkownika (okienko wyświetlacza OLED i przyciski funkcyjne) oraz otwór (w bocznej ścianie)

**Rysunek 6. Widok 3D obudowy urządzenia powerTracker**

REKLAMA

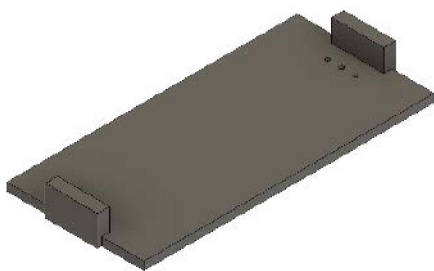
Hurtownia elementów elektronicznych "AKSOTRONIK" zaprasza do swojego sklepu internetowego  
Zaloguj się i kupuj ON-LINE na naszej stronie:  
**WWW.AKSOTRONIK.COM.PL**

**Aksotronik**  
ELEMENTY ELEKTRONICZNE

- Magnesy neodymowe oraz ferrytowe  
Ceny od 0.10zł
- Przełączniki klawiszowe wodoszczelne/pyłoszczelne  
Ceny od 2.40zł
- Druty oporowe od 0.16 do 0.81mm  
Ceny od 5.70zł
- Prowadniki do przewodów  
Ceny od 11.80zł
- Kostki elektryczne zaciskowe  
Ceny od 0.22zł
- Szczotki węglowe do elektronarzędzi  
Ceny od 2.60zł/kpl
- Przełączniki do elektronarzędzi zwykłe i elektromagnetyczne  
Ceny od 7.00zł
- Złącza hermetyczne Superseal  
Ceny od 1.10zł/kpl
- Podkładki/organizery  
Ceny od 0.95zł
- Zestawy śrubek M2, M3 z nakrętkami i podkładkami  
Ceny od 2.50zł

Uwaga!! Powyższe ceny dotyczą zakupów minimalnych ilości hurtowych, poprzez nasz sklep internetowy.  
W swojej ofercie posiadamy m.in.: półprzewodniki (diody, układy scalone, tranzystory, triaki, elementy optoelektryczne), elementy dystansowe, złącza, przełączniki, elementy akustyczne, rezystory, kondensatory, kwarcy, podstawki, moduły Arduino  
Zapraszamy do kontaktu: **INFO@aksotronik.com.pl**, tel: (22) 783-20-51



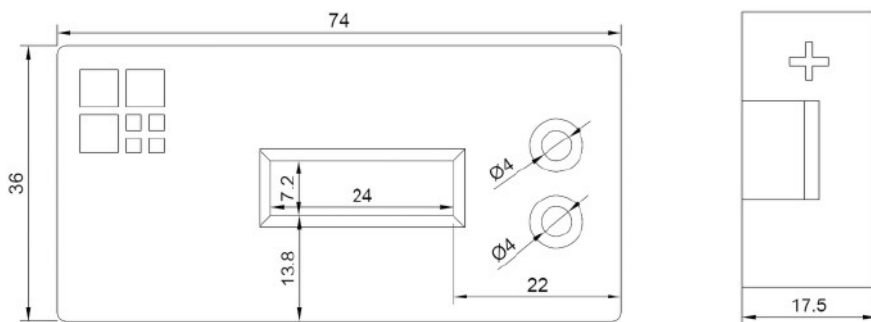


Rysunek 7. Widok 3D klapki stanowiącej drugi element obudowy urządzenia powerTracker

na gniazdo karty SD – a także części dolnej pełniącej funkcję klapki, za pomocą której zamykamy obudowę od dołu. Wygląd 3D (od wewnętrznej strony obudowy) klapki, stanowiącej drugi element obudowy, pokazano na **rysunku 7**.

Z kolei **rysunku 8** pokazano widok 2D wspomnianej obudowy z zaznaczeniem wymiarów kluczowych krawędzi.

Stosowne pliki projektu obudowy (do wydrukowania na drukarce 3D) znajdują się w powiązanych z artykułem materiałach dodatkowych, dostępnych na stronie ep.com.pl. Dodatkowo podpowiem, że – jeśli nie dysponujecie odpowiednim urządzeniem umożliwiającym wydrukowanie obudowy według załączonych plików – z powodzeniem możecie zlecić takie zadanie firmie świadczącej usługi druku 3D. Wybierając tego rodzaju rozwiązanie, decydujcie się na wydruk w technologii MJF (*Multi Jet Fusion*) lub STS (*Selective Laser Sintering*). Pierwsza z nich polega na druku 3D ze sproszkowanych tworzyw sztucznych (poliamidów), poprzez selektywne natryskiwanie na nie lepiszczą (które skleja ze sobą poszczególne warstwy modelu), a następnie zgrzewanie w wysokiej temperaturze, co powoduje ich trwałe zespolenie się. Atutem wydruków 3D tego rodzaju jest wysoka wytrzymałość mechaniczna produkowanych części. Uzyskuje się ją dzięki jednolitej strukturze, która ma 100-procentowe wypełnienie. Druga z polecanych przeze mnie metod, SLS, to przyrostowa technika druku przestrzennego, polegająca na warstwowym spiekaniu laserem proszku polimerowego, który stopniowo tworzy trójwymiarowy



Rysunek 8. Widok 2D obudowy urządzenia powerTracker z zaznaczeniem wymiarów kluczowych krawędzi

wydruk. Detale realizowane w tej technologii charakteryzują się bardzo dobrym przyleganiem warstw i świetnymi właściwościami mechanicznymi, szczególnie odpornością na zgniatanie czy ścieranie. Ponadto cechują się one ogromną dokładnością, dlatego można wykorzystywać je nawet w bardzo precyzyjnych zastosowaniach. Szczepnie polecam tego rodzaju rozwiązanie, dodając, że zamieszczona na fotografii tytułowej obudowa została wydrukowana właśnie w tej technologii (STS), a koszt całej usługi (wydruku 2 elementów obudowy) wyniósł około 5 \$. Prawda, że tanio? I zupełnie na sam koniec słowo komentarza na temat programowania mikrokontrolera – należy wykonać je przy użyciu nowoczesnego interfejsu programowania UPDI, w który wyposażono nasz układ. Istnieją co najmniej 2 sposoby realizacji tego zadania: albo przy użyciu programatora firmy Microchip o nazwie MPLAB SNAP – albo programatora UPDI mojego autorstwa o nazwie sUPDI, który był tematem artykułu z numeru 4/2021 naszego miesięcznika. Co ciekawe, jest to możliwe bez żadnych specjalnych przygotowań mikrokontrolera, gdyż wyprowadzenie UPDI/RESET układu standardowo ustawione jest za pomocą bitów konfiguracyjnych (fuse-bitów) właśnie jako linia interfejsu UPDI. Niemniej jednak uważaj! Czytelnik zauważy zapewne, iż w aplikacji urządzenia powerTracker do tego wyprowadzenia podłączone jest jednocześnie wyjście ALERT układu INA226, służące do alarmowania po przekroczeniu wartości progowych, będących przedmiotem pomiaru. O ile takiej

funkcjonalności nie wyklucza standardowe ustawienie fuse-bitów mikrokontrolera, gdyż wyprowadzenie w trybie UPDI może być używane (z pewnymi ograniczeniami) jako zwykłe wejście (z wewnętrznym podciąganiem), o tyle sama obecność podłączenia tego pinu z wyjściem ALERT przetwornika INA226 może uniemożliwić programowanie mikrokontrolera (zwłaszcza gdy jest w stanie alarmowania – stanie niskim). Oznacza to, że w przypadku tego rodzaju problemów, wyprowadzenie ALERT układu INA226 należy odłączyć od pinu UPDI podczas programowania mikrokontrolera. Z drugiej strony: mogłoby się również wydawać, że podczas normalnej pracy układu stan wyprowadzenia ALERT wspomnianego przetwornika mógłby doprowadzić do wprowadzenia mikrokontrolera w tryb programowania. Tak, sytuacja taka jest teoretycznie możliwa, lecz nie może mieć miejsca w przypadku naszego urządzenia. Aby wejść w tryb programowania, należy, po spełnieniu pewnych timingów wejściowych, przesłać bajt synchronizacji o wartości 0x55, czego układ INA226 z pewnością wykonać nie może. Osoby szczególnie wrażliwe na tego typu łączenia funkcjonalności mogą, po zaprogramowaniu mikrokontrolera, przestawić (za pomocą zaprogramowania fuse-bitu FUSE\_SYSCFG0) funkcję wyprowadzenia UPDI/RESET jako normalnego portu I/O, jednak po wykonaniu takiej operacji wejście w tryb programowania wymagać będzie programatora HV (wysokonapięciowego).

**Robert Wołgajew, EP**

REKLAMA

Świat projektantów i programistów dla elektroniki w nowej odświeżonej. Odwiedź wiecznie młody

**ELPORTAL.pl**

Znajdziesz nas również na Facebooku: [facebook.com/ElportalPL](https://facebook.com/ElportalPL)

**Podstawowe parametry:**

- dwa niezależnie sterowane kanały,
- prąd szczytowy uzwojeń: 3 A/kanał,
- napięcie zasilania silników: 3...30 V (niezależne w obydwu kanałach),
- możliwość sterowania kierunkiem i prędkością obrotów,
- możliwość monitorowania prądu silników,
- konstrukcja na bazie scalonych sterowników petnomostkowych MAX22202,
- wbudowana przetwornica DC/DC typu buck z opcją wyboru źródła zasilania logiki (napięcie zasilania kanału 1 lub 2).

\* **Uwaga!** Elektroniczne zestawy do samodzielnego montażu. Wymagana umiejętność lutowania! Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wylutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wylutowane w płytkę PCB),
  - wersja [A] – płytkę drukowaną bez elementów i dokumentacji.
- Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- wersja [A+] – płytkę drukowaną [A] + zaprogramowany układ [UK] i dokumentacja,
  - wersja [UK] – zaprogramowany układ.

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

- AVT6023 Nakładka Ethernet PoE do AVTDuino (EP 2/2024)
- AVT5850 Płytki bazowa dla Arduino Nano Every (EP 3/2021)
- AVT5819 Płytki bazowa dla Arduino MKR (EP 11/2020)
- AVT5777 Moduł interfejsu ethernet dla Arduino MKR Zero (EP 6/2020)
- AVT5738 Wyświetlacz 7-segmentowy dla Arduino MKR Zero (EP 1/2020)
- Sterownik silników prądu stałego dla Arduino MKR (EP 3/2019)
- AVT5636 Płytki edukacyjna dla Arduino (EP 5/2018)
- AVT1795 AVTDuino Battery Shield (EP 3/2014)
- AVT1722 AVTDuino miniLCD – miniaturowy panel operatora dla Arduino (EP 1/2013)
- AVT1686 AVTRelDuino Shield. Moduł wykonawczy dla Arduino (EP 8/2012)

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik PDF! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! <http://sklep.avt.pl>

W przypadku braku dostępności na stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: [kity@avt.pl](mailto:kity@avt.pl)

W ofercie AVT\*  
**AVT6028**

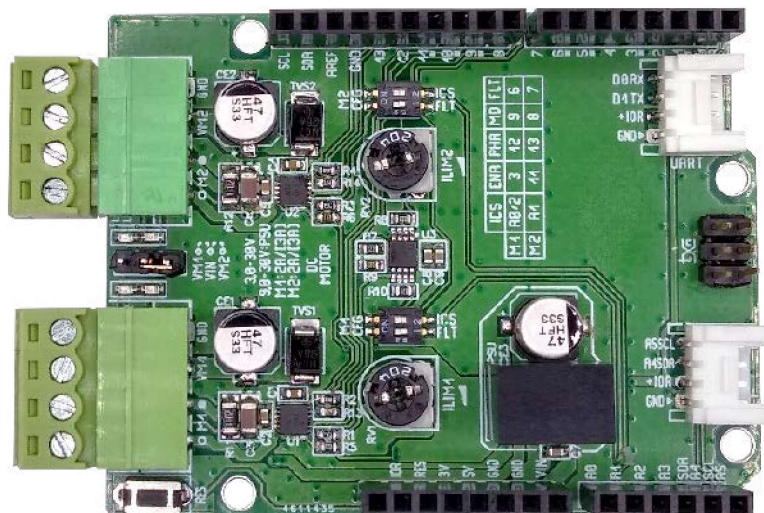
# Sterownik silników do AVTDuino UNO R4

Przedstawiona konstrukcja kontynuuje cykl odświeżający nakładki Arduino, pozwalający na korzystanie z całego potencjału wersji R4 popularnej płytki bazowej. Opisany moduł zastępuje leciwe sterowniki silników oparte na układach L293 oraz L298 nowoczesnymi sterownikami MAX22202, zachowując możliwie wysoki stopień kompatybilności wstecznej.

Nakładka umożliwia dwukierunkowe sterowanie sygnałem PWM dwóch silników prądu stałego z niezależną regulacją prędkości. Prąd szczytowy uzwojeń może sięgnąć maksymalnie 3 A przy zasilaniu każdego silnika z osobnego źródła napięcia 3...30 VDC. Wbudowana opcjonalna przetwornica obniżająca ułatwia zasilanie płytki bazowej, gdy napięcie jednego z silników mieści się w zakresie 9...30 V. Zastosowane układy sterowników zapewniają płynną regulację prądu uzwojeń, z wartością ustaloną potencjometrem montażowym; mają ponadto funkcję monitorowania prądów uzwojeń oraz sygnalizację stanów awaryjnych.

Nakładka wyposażona jest w dwa nowoczesne, pełnomostkowe układy sterownika silnika prądu stałego typu MAX22202 (Analog Devices), którego budowę przedstawiono na rysunku 1.

Układ MAX22202 zawiera wszystkie bloki niezbędne do sterowania silnikiem prądu stałego, a jego aplikacja ogranicza się do zaledwie kilku elementów zewnętrznych. Wbudowany mostek MOSFET charakteryzuje się niewielką rezystancją w stanie przewodzenia  $R_{ds(on)} \sim 0,3 \Omega$ , co zmniejsza straty i umożliwia



pracę przy niskich napięciach zasilania, już od 3 V, bez zmniejszenia wydajności prądowej. MAX22202 ma wbudowany układ pomiaru prądu silnika, niewymagający zewnętrznego rezystora pomiarowego – dzięki temu jego aplikacja jest uproszczona, a straty mocy zminimalizowane. Układ pracuje w dwóch trybach: pierwszy to włączona regulacja prądu silnika, przy którym maksymalny prąd uzwojeń ustalany jest rezystancją podłączoną do wprowadzenia ILIM, drugi to praca z wyłączoną regulacją, gdy do wyprowadzenia ILIM podłączony jest rezystor 14 k $\Omega$ , a aktywnie pozostaje tylko ograniczenie prądu maksymalnego 3,5 A. Regulacja prądu silnika ułatwia np. rozruch napędu bez przeciążeń prądowych i zbędnych obciążeń mechanicznych. W każdym trybie możliwa jest regulacja PWM. Wartość rezystora podłączonego do wyprowadzenia ILIM ustalającego prąd silnika  $I_{trip}$  wyznacza się ze wzoru:

$$I_{trip} [A] = K_{ilim} / R_{lim} [k\Omega]$$

gdzie  $K_{ilim} = \sim 50 \text{ KV}$  ( $R_{lim} : 18...250 \text{ k}\Omega$ )

Jeżeli w danej aplikacji przewidywane jest sterowanie silnika w zamkniętej pętli sprzężenia zwrotnego lub konieczna jest informacja o prądzie uzwojeń, to wyprowadzenie ISEN udostępni potrzebny sygnał

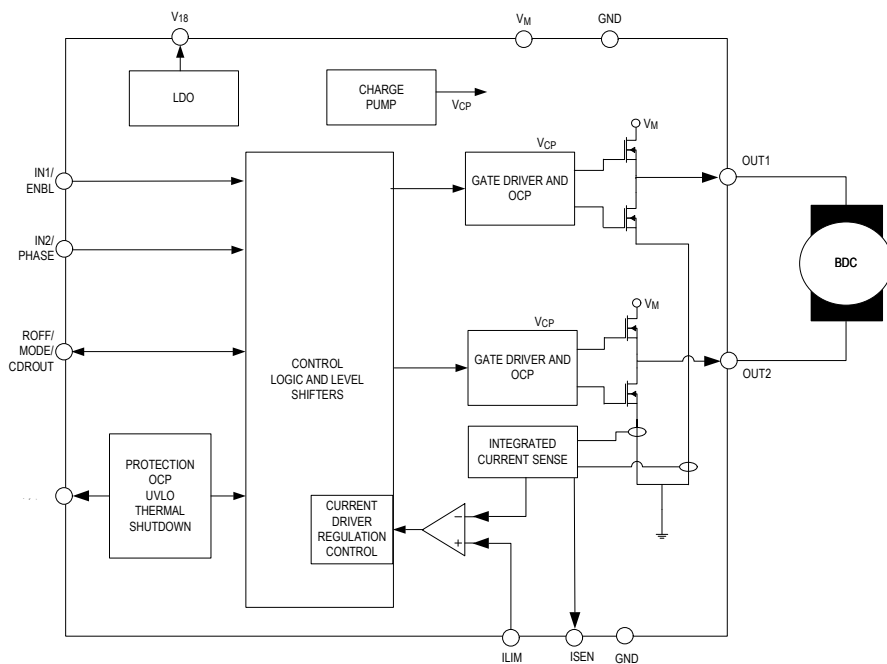
REKLAMA

## LASEROWE SZABLONY DO MONTAŻU SMT

Materiał: stal nierdzewna CrNi  
Zakres grubości blach: 0,020–1,000 mm  
Wycinamy również detale o dowolnych kształtach



LASTENIC LASER & ELECTRONICS sp. z o.o.  
58-100 Świdnica, ul. Husarska 5  
tel. 74 851 48 77, 697 977 732  
[www.lastenic.com](http://www.lastenic.com) [info@lastenic.com](mailto:info@lastenic.com)



Rysunek 1. Budowa wewnętrzna MAX22202 (za notą Analog Devices)

monitorujący. Podłączony do niego rezystor obciąża wbudowane źródło prądowe, a uzyskiwane napięcie wyjściowe opisane jest wzorem:

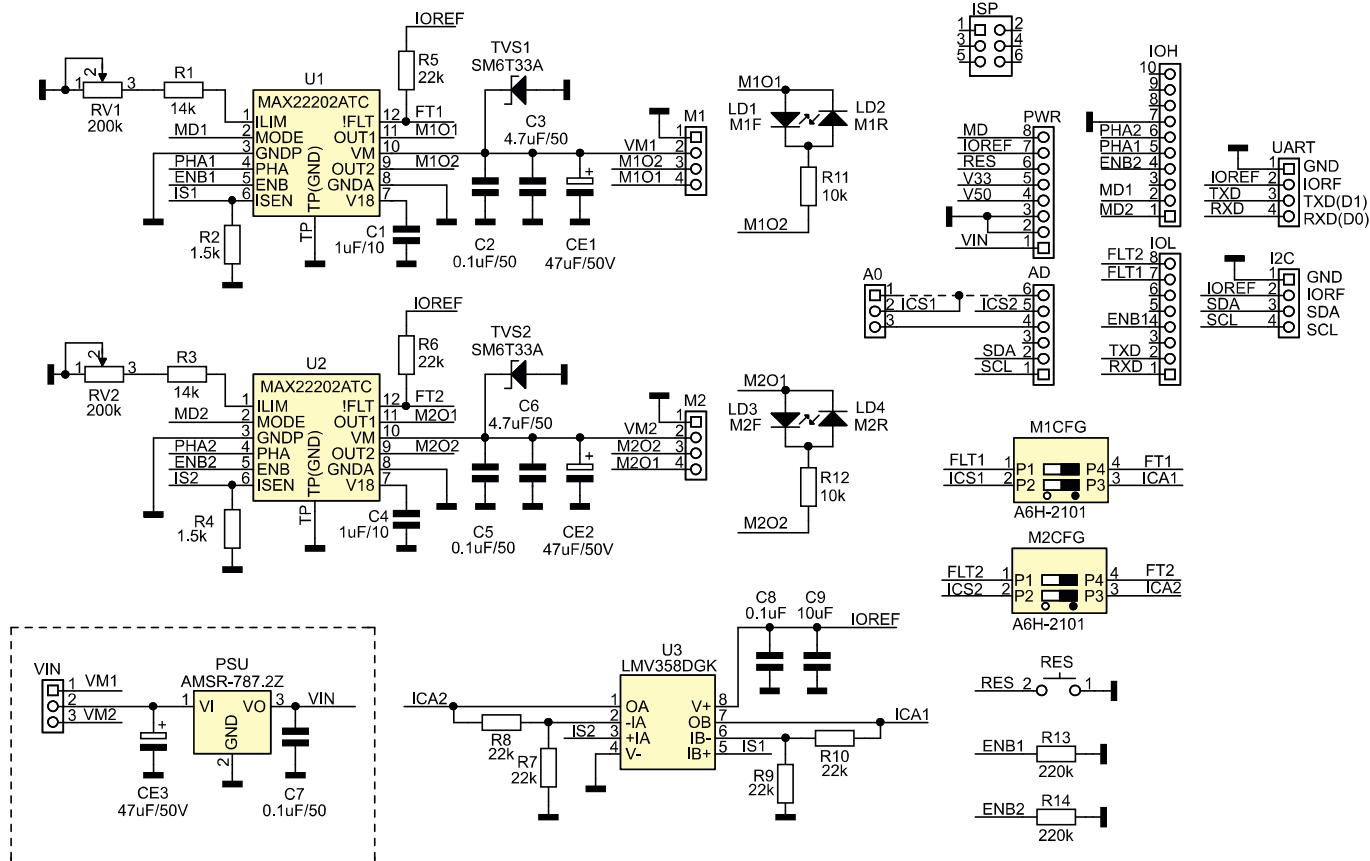
$$R_{isen} [\Omega] = K_{isen} \cdot V_{fs} [V] / I_{max} [A]$$

gdzie  $K_{isen} \approx 5 \text{ kA/A}$

Stany awaryjne – takie jak przeciążenie, przegrzanie lub zbyt niskie napięcie zasilania – sygnalizowane są stanem niskim na wyprowadzeniu FAULT.

Za sterowanie silnikiem odpowiadają sygnały ENABLE, PHASE(DIR) oraz MODE, zgodnie z tabelą 1. Jeżeli stan niski na wyprowadzeniu MODE trwa dłużej niż (1,5...3,5 ms), układ przechodzi w stan obniżonego poboru mocy. W przypadku włączonej regulacji prądu silnika i przekroczenia jej nastawy, sterowanie mostkiem przełączane jest na wewnętrzne, ograniczające prąd uzwojenia niezależnie od stanu wyprowadzeń.

Schemat nakładki przedstawiono na rysunku 2. Układy U1, U2 (MAX22202)



Rysunek 2. Schemat nakładki

**Wykaz elementów:**

**Rezystory:** (1%)

- R1, R3: 14 kΩ (SMD 0603)
- R2, R4: 1,5 kΩ (SMD 0603)
- R5...R10: 22 kΩ (SMD 0603)
- R11, R12: 10 kΩ (SMD 1206)
- R13, R14: 220 kΩ (SMD 0603)

**Kondensatory:**

- C1, C4: 1 μF/10 V (SMD 0603, X7R)
- C2, C5, C7: 100 nF/50 V (SMD 0603, X7R)
- C3, C6: 4,7 μF/50 V (SMD 1206, X7R)
- C8: 100 nF/50 V (SMD 0603, X7R)

- C9: 10 μF/10 V (SMD 0603, X7R)
- CE1...CE3: 47 μF/50 V elektrolityczny, typ EEEFT1H470AP

**Półprzewodniki:**

- LD1, LD3: dioda LED czerwona (SMD 0603)
- LD2, LD4: dioda LED zielona (SMD 0603)
- TVS1, TVS2: tranzil SM6T33A (SMB)
- U1, U2: MAX22202ATC (TDFN12)
- U3: LMV358DGK (VSSOP8\_065)

**Pozostałe:**

- AD: złącze szpilkowe 6 pin 2,54 mm (13...15 mm)

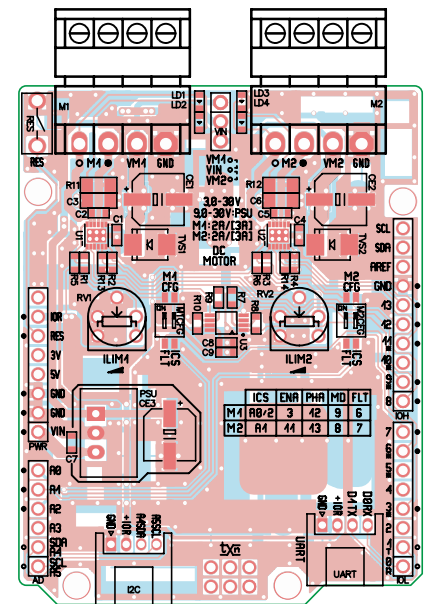
- I<sup>2</sup>C, UART: złącze Grove kątowe
- IOH: złącze szpilkowe 10 pin, 2,54 mm (13...15 mm)
- IOL, PWR: złącze szpilkowe 8 pin, 2,54 mm (13...15 mm)
- ISP: złącze szpilkowe 2x3 pin, 2,54 mm (13...15 mm) (opcja)
- M1, M2: złącze śrubowe kompletne 4 pin 3,81 mm kątowe
- M1CFG, M2CFG: przełącznik DIP switch, 2 sekcje, typ A6H-2101
- PSU: przetwornica AMSR-787-2Z (opcja)
- VIN: złącze SIP3 + zwora (opcja)
- RES: przycisk TACT-switch 6x3 mm
- RV1, RV2: potencjometr montażowy 200 kΩ



pracują w typowej aplikacji. Każdy z silników podłączony jest wraz z napięciem zasilania do odpowiadającego mu złącza M1, M2. Odsprężanie zasilania zapewniają kondensatory C2, C3, CE1 i C5, C6, CE2, zaś transile TVS1, TVS2 zabezpieczają drivery przed skutkami przepięć. Pojemności filtrujące mają minimalne wartości, niezbędne do zapewnienia poprawnej pracy U1, U2 (każdorazowo należy zapewnić odpowiednią filtrację i przekroje przewodów zasilających dobranych pod kątem wymagań współpracujących silników). Regulacja prądu silnika odbywa się za pomocą potencjometrów RV1, RV2. Ustawienie ślizgaczy w maksymalnym położeniu wyłącza regulację, ustalając rezystancję widzianą przez wyprowadzenie ILIM na R1, R3=14 kΩ. Sygnał monitorujący prąd silnika, dostępny na wyprowadzeniu ILIM, przetwarzany jest na napięcie i dodatkowo buforowany przez wzmacniacz U3 o wzmacnieniu ustalonym na 2 V/V. Maksymalny mierzony prąd to 3,5 A, co odpowiada napięciu 2 V na wyprowadzeniach A0 lub A2 i A1. Zakres prądu i napięcia może zostać zoptymalizowany poprzez dobór rezystorów R2, R10 i R4, R8 odpowiednio do każdego z silników – należy pamiętać tylko, by nie przekraczać zakresu napięcia wejściowego ADC (zależnego od użytej płytki bazowej). Bezpiecznie jest pracować w zakresie 0...3 V, ponieważ gwarantuje zgodność z płytkami zasilanymi napięciami zarówno 3,3 V, jak i 5 V. Zwora A0 (zalutowana domyślnie w położeniu A0) umożliwia zmianę wyprowadzenia monitorującego prąd silnika M1 z A0 na A2, co jest istotne w aplikacjach, w których chcemy wykorzystać wyprowadzony na A0 przetwornik D/A w UNO R4. Sygnał monitorujący prąd silnika nie podlega w układach U1 i U2 żadnej obróbce oprócz skalowania zakresu – o stosowne przeliczenie należy zatem zadbać we własnym zakresie, zwłaszcza jeśli sterujemy silnikiem sygnałem PWM. Kiedy sygnał monitorujący prąd silników nie jest niezbędny w aplikacji, może zostać całkowicie odłączony od wyprowadzeń A0(2), A1 przy użyciu

przełączników DIP M1CFG, M2CFG, a linie te zostaną uwolnione do użycia w innych celach. Stan awaryjny driverów: FLT1, FLT2, dostępny jest na wyprowadzeniach D6, D7 i także może zostać od nich odłączony przełącznikami DIP M1CFG, M2CFG. Sterowanie silnikami sprowadza się do trzech sygnałów: ENB1/2 (złączającego silnik z możliwą regulacją PWM), PHA1/2 (odpowiadającego za zmianę kierunku) oraz MD1/2 (odpowiadającego za tryb pracy). Przypisanie sygnałów jest zgodne z Motor Shield. Diody LD1...LD4 – podłączone bezpośrednio na wyjściach OUT układów U1 i U2 – sygnalizują wysterowanie silnika, dzięki czemu możliwa jest też obserwacja napięcia indukowanego w silniku przy braku wysterowania uzwojenia (przejście do pracy prądnicowej) i kontrolowanie rzeczywistego stanu wyjść mostka.

W celu ułatwienia aplikacji nakładki, na złącza Grove wyprowadzono magistralę I<sup>2</sup>C (wyprowadzenia A4, A5) oraz UART (D0, D1), umożliwiającą podłączenie czujników zasilanych napięciem IOREF. Opcjonalnie układ uzupełniono przetwornicą obniżającą PSU typu AMSR-787.2Z, pozwalającą na zasilanie płytki bazowej z jednego z napięć zasilających silniki, doprowadzonych do złącza VIN – połączenie współpracującej z nim zwory ustala wybór napięcia VM1 lub VM2. Przetwornica obniża wybrane napięcie do 7,2 V (obciążalność maksymalna 500 mA). Do poprawnej pracy konwertera DC/DC wymagane jest napięcie wejściowe 9...30 V, co nieznacznie ogranicza możliwości driverów, ale upraszcza aplikację w przypadku, gdy podłączamy silniki do typowego napięcia z przedziału 9...24 V, np. z pakietu akumulatorów. Gdy wymagane jest sterowanie silnikami niskonapięciowymi 3...9 V, napięcie do płytki bazowej musi być doprowadzone niezależnie. W przypadku AVTduino UNO R4 Plus, aby wykorzystać zasilanie z napięcia silników, należy założyć na płytkę bazową zworę VIN; w przypadku innych płytek Arduino trzeba koniecznie sprawdzić dopuszczalny zakres zasilania VIN.



Rysunek 3. Rozmieszczenie elementów na płycie drukowanej

Układ zmontowany jest na dwustronnej płycie drukowanej zgodnej z Arduino Shield Rev3. Rozmieszczenie elementów przedstawiono na rysunku 3. Sposób montażu nie wymaga opisu, montaż zwory VIN i przetwornicy PSU jest opcjonalny (zgodnie z uwagami w tekście). W zależności od przewidywanego zastosowania, przedłużane złącza szpilkowe PWR, AD, IOL, IOH, ISP (wysokość 13...15 mm) można zastąpić „stackowalnymi” złączami Arduino, umożliwiającymi montowanie modułów w kanapki.

Prosty szkic z listingu 1 ułatwia przetestowanie sterowania silnikami przy użyciu PWM i zmiany kierunku. W przypadku zastosowania szkieletów z Arduino Motor Shield należy zmodyfikować funkcję obsługującą wyprowadzenie BRAKE, która w AVTduino UNO\_R4\_DCMotor Shield pełni funkcję MODE i konieczna jest jej negacja.

Jeżeli wszystko działa poprawnie, moduł może zostać użyty we własnej aplikacji.

Adam Tatuś, EP

REKLAMA

Nie przegap marcowego wydania  
„Elektroniki dla Wszystkich”

przejrzyj i kupisz na  
[www.ulubionykiosk.pl](http://www.ulubionykiosk.pl)





**Podstawowe parametry:**

- Dwa wyjścia stałonapięciowe: 12 V/2 A oraz 5 V/2,5 A
- Łączna moc wyjściowa: 24 W (ciągła)/30 W (szczytowa)
- Sprawność głównej przetwornicy 12 V: 89%
- Możliwa regulacja napięcia pierwszego stopnia układu w zakresie 10,8 V..12,8 V
- Konstrukcja oparta na module Ag5712
- Zdublowane gniazda RJ45, zapewniające łatwą rozbudowę sieci przy jednoczesnym korzystaniu z funkcjonalności PoE i transmisji LAN
- Wbudowane wskaźniki obecności napięcia na obu wyjściach (2xLED SMD)

\* **Uwaga!** Elektroniczne zestawy do samodzielnego montażu. Wymagana umiejętność lutowania! Podstawową wersją zestawu jest wersja [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wylutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

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

- AVT5977 Zasilacz do płytek stykowych ze złączem USB-C (EP 10/2023)
- AVT5963 Stabilizator napięcia symetrycznego z regulacją współbieżną (EP 3/2023)
- Zasilacz warsztatowy, część 1 i 2 (EP 12/2022, 01/2023)
- Modułowy zasilacz warsztatowy (EP 5/2022)
- Regulowany zasilacz warsztatowy – RPS-02 (EP 4/2022)
- AVT5915 Zasilacz 5 V/1 A z szerokim zakresem napięć wyjściowych (EP 1/2022)
- AVT5908 Beztransformatorowy impulsowy zasilacz sieciowy (EP 12/2021)
- AVT5872 Regulowany zamiennik stabilizatora 78xx (EP 7/2021)
- AVT1990 Regulowany zasilacz do płytek stykowych (EP 8/2018)
- Precyzyjny regulowany zasilacz stabilizowany (EP 2/2018)

- wersja [C] – zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wylutowane w płytce PCB),
  - wersja [A] – płytkę drukowaną bez elementów i dokumentacji.
- Kity, w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- wersja [A+] – płytkę drukowaną [A] + zaprogramowany układ
  - [UK] i dokumentacja,
  - wersja [UK] – zaprogramowany układ.

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik PDF! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! <http://sklep.avt.pl>

W przypadku braku dostępności na stronie sklepu osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: [kity@avt.pl](mailto:kity@avt.pl).

W ofercie AVT\*  
**AVT6029**

# Zasilacz PoE 5/12 V 30 W

Zasilacz umożliwia zastosowanie standardu 802.3at (PoE+) w celu uzupełnienia lub rozbudowy urządzeń korzystających z interfejsu Ethernet, a niewyposażonych w funkcję zasilania. Dostarcza napięcia 5 V/2,5 A i 12 V/2 A o sumarycznej mocy 24 W (szczytowo 30 W), co poszerza zakres zastosowań i ułatwia dostarczanie energii do urządzeń zawierających logikę 5-woltową oraz obwody sterowane, np. jak przekaźniki zasilane z 12 V lub dyski HDD. Zasilacz wyposażony został w gniazda obciążeniowe do transmisji Ethernet100M, co rozwiązuje jednocześnie dwa problemy: z zasilaniem i i transmisją danych do urządzenia.

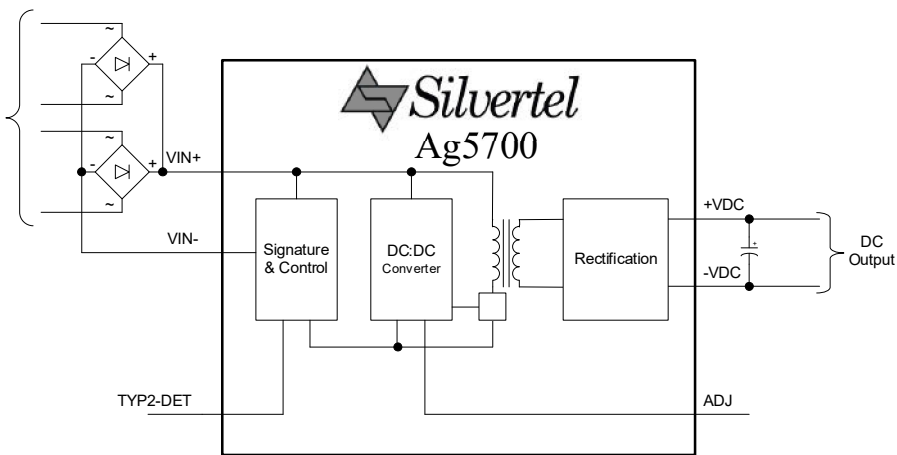


Układ bazuje na module Ag5712 Silvertel, którego budowę wewnętrzną przedstawiono na rysunku 1.

Ag5712 pozwala spełnić wymogi IEEE802.3 PoE, przy utrzymaniu pełnej kompatybilności z IEEE802.3bt, IEEE802.3at i IEEE802.3af. Ag5712 jest wstępnie skonfigurowany jako urządzenie typu 2 klasy 4, umożliwia pobieranie do 30 W mocy z PSE, przy nominalnym napięciu wyjściowym 12 V i sprawności do 89%. Wbudowana przetwornica zabezpieczona została przed przeciążeniem oraz zwarcieniem wyjścia, a także przegrzaniem.

Schemat zasilacza przedstawiono na rysunku 2. Zasilanie z routera, switcha lub iniektora PoE+ wraz z sygnałami transmisji doprowadzone jest typowym okablowaniem Cat5 do gniazda ETHPoE. Same przebiegi cyfrowe nie podlegają żadnym modyfikacjom i są bezpośrednio wprowadzone na gniazdo ETH, które należy połączyć z interfejsem sieciowym urządzenia

**UWAGA:** w module przetwornicy występują napięcia do 60 V, należy zachować szczególną ostrożność podczas uruchamiania układu. W celu zwiększenia bezpieczeństwa polecamy dodatkowo zabezpieczyć moduł przekładką lub taśmą izolacyjną.



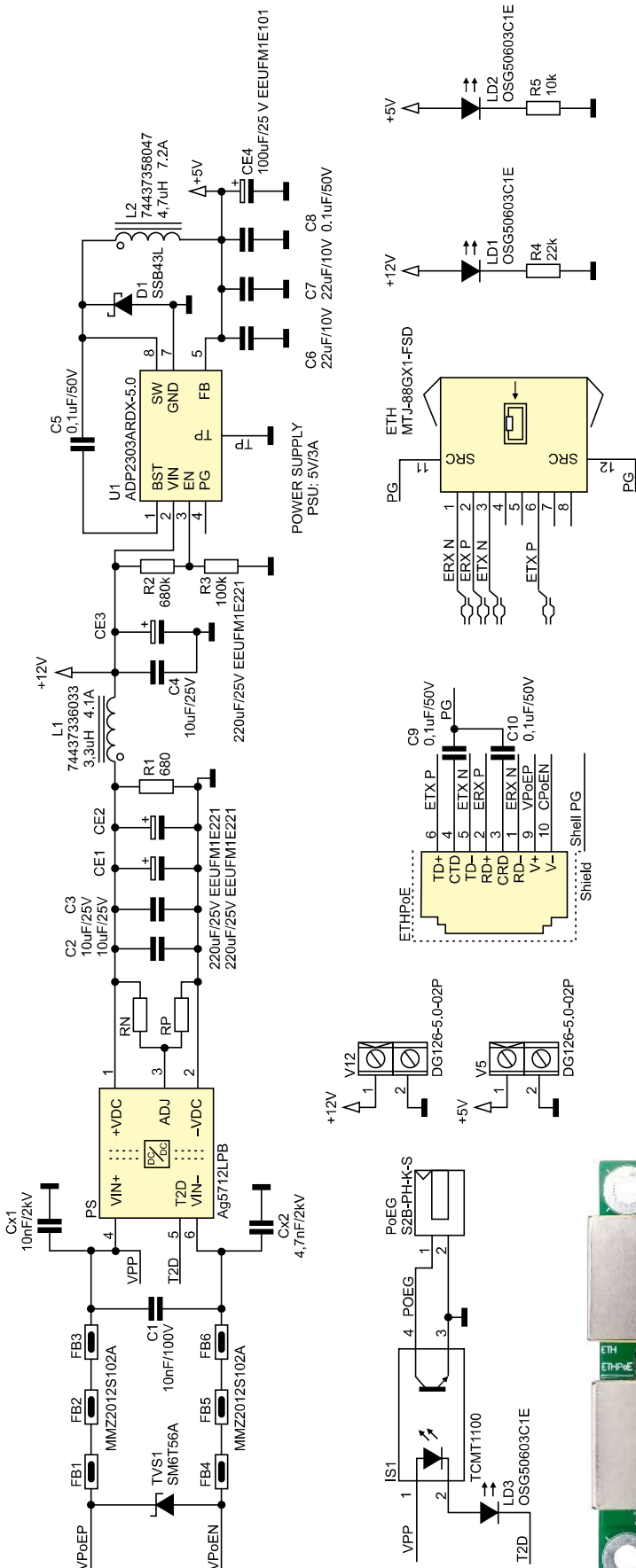
Rysunek 1. Schemat blokowy Ag5712 (za notą Silvertel)



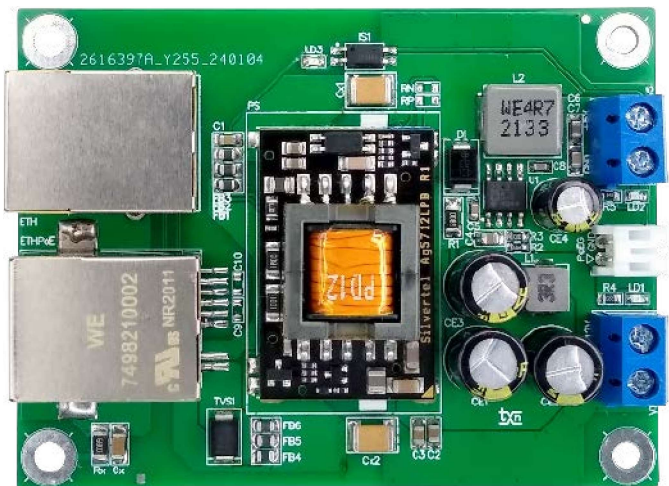
(lub pozostawić niepodłączone, jeżeli używamy tylko trybu zasilania). Modułowe gniazdo ETHPoE typu WE 7498210002 wyposażone jest w transformatory separacyjne

dla kanałów transmisji RX/TX oraz dwa mostki prostownicze umożliwiające zastosowanie obu schematów transmisji zasilania, zarówno przez pary sygnałowe, jak i przez

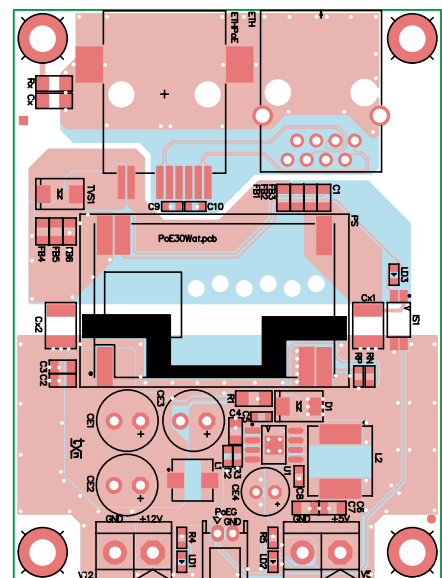
pary niewykorzystane w interfejsie 100M. Zasilanie z PoE+ – po zabezpieczeniu przez transil TVS1 i dodatkowej filtracji zakłóceń poprzez FB1...FB6 – doprowadzone zostaje do modułu PS. Wyjście T2D służy do sygnalizacji typu współpracującego PSE. Jeżeli moduł podłączony został do PSE zgodnego z PoE+, T2D ustawiane jest w stan niski (co sygnalizuje tym samym dostępność pełnej mocy modułu). Transceptor IS1 separuje sygnał od potencjału pierwotnej strony przetwornicy, zaś dioda LD3 sygnalizuje zgodność z PoE+. Jeżeli PSE jest niezgodne z PoE+, wówczas wyjście T2D ustawiane zostaje w stan wysoki, jednocześnie informując współpracujący układ o ograniczeniu dostępnej mocy do 12 W. Sygnał POEG dostępny jest na złączu PoEG. Rezystory RN, RP służą do korekcji napięcia wyjściowego przetwornicy, zgodnie z notą aplikacyjną. Włutowując rezystor RN=68 kΩ, obniżamy napięcie przetwornicy do 10,8 V, natomiast włutowując zwrót w miejsce RP – podnosimy napięcie wyjściowe do 12,8 V.



Rysunek 2. Schemat ideowy zasilacza



Fotografia 1. Zmontowany moduł



Rysunek 3. Rozmieszczenie elementów modułu