



# Ampera – miniaturowy miernik prądu stałego

Prezentujemy prosty, ale efektywny amperomierz prądu stałego, który ma możliwość alarmowania po przekroczeniu konfigurowalnego prądu. Projekt o nazwie Ampera.

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

**W ofercie AVT\* AVT-5766**

**Podstawowe parametry:**

- zakres pomiarowy: 0...8 A,
- praca w systemach o napięciu szyny zasilającej 0...36 V,
- rozdzielczość pomiaru: 1 mA,
- dokładność pomiaru: 1% (zależna od klasy dokładności zastosowanego bocznika)
- zakres ustawień dla funkcji alarmowania: 0...10 A,
- rozdzielczość ustawień dla funkcji alarmowania: 1 mA,
- częstotliwość odświeżania ekranu: 10 razy/s,
- zasilanie: 4,5...10 V/10 mA.

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

- Projekt 233 3-fazowy woltomierz TRMS (EP 10/2017)
- Projekt 232 Miliamperomierz TRMS (EP 9/2017)
- AVT-5507 Miernik UIPTR (EP 7/2015)
- AVT-5399 Dwukanałowy multimetr panelowy (EP 6/2013)
- AVT-5386 Podwójny woltomierz i amperomierz (EP 3/2013)
- AVT-5383 Miernik tablicowy UIPT (EP 2/2013)

**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 wzlutować w dołączoną płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:

- wersja [C] - zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wzlutowane w płytkę PCB)
- wersja [A] - płytkę drukowaną bez elementów i dokumentacji Kitu w których występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
- wersja [Ae] - 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 <http://sklep.avt.pl>, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: [kity@avt.pl](mailto:kity@avt.pl).

W urządzeniu zastosowano specjalizowany układ przeznaczony do pomiaru prądu, napięcia i mocy urządzeń zasilanych napięciem stałym. Jest to element dość wyjątkowy, warto choćby skrótowo zaznajomić się z jego specyfikacją. Układ INA226, produkcji firmy Texas Instruments, jest specjalizowanym, bardzo dokładnym, 16-bitowym, różnicowym przetwornikiem pomiarowym ADC przeznaczonym do zastosowania w układach pomiaru prądu i mocy z wykorzystaniem bocznika rezystancyjnego. Układ ten wyróżnia się następującymi, wybranymi cechami użytkowymi:

- szeroły 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 konfiguracji 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 zadanego poziomu prądu, napięcia szyny zasilającej odbiornik bądź mocy pobieranej przez odbiornik.

Układ INA226 idealnie wpisuje się w wymagania naszej aplikacji, oferując niespotykaną dotąd funkcjonalność i dokładność pomiarów. Schemat blokowy tego peryferium pokazano na **rysunku 1**.

## Budowa i działanie

Schemat ideowy urządzenia znajduje się na **rysunku 2**. Zaprojektowano bardzo prosty system mikroprocesorowy, którego sercem jest niewielki mikrokontroler firmy Microchip (dawniej Atmel) typu ATtiny44, odpowiedzialny za programową implementację interfejsu I<sup>2</sup>C, przy użyciu którego mikrokontroler realizuje obsługę układu INA226 oraz obsługę niewielkiego, acz bardzo efektywnego wyświetlacza OLED o rozdzielczości 128×32 piksele stanowiącego element

**Wykaz elementów:**

**Rezystory:**

- R1: pomiarowy 10 mΩ 1 W 1% SMD2010
- R2, R3: 4,7 kΩ SMD 0805

**Kondensatory:**

- C1..C3: ceramiczny X7R 100 nF SMD 0805
- C4: ceramiczny X7R 1 μF SMD 0805

**Półprzewodniki:**

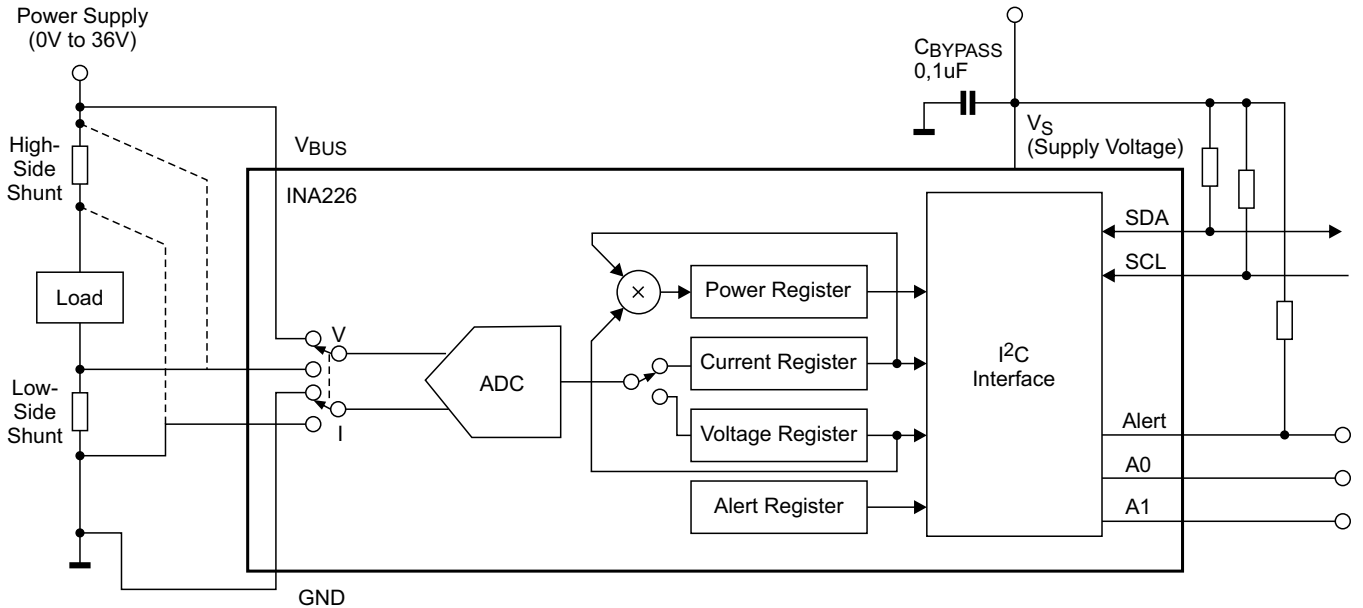
- U1: ATtiny44 (SOIC14)
  - U2: INA226A (MSOP-10)
  - U3: TS9011SCX (SOT23)
- OLED: wyświetlacz OLED 128×32 px, 0,91", SSD1306, magistrała I<sup>2</sup>C, wymiary 38×12 mm

**Pozostałe:**

- PLUS, MINUS: mikroprzełącznik SMD typu DTSM31NB
- BUZZ: przetwornik piezoelektryczny typu LPT9018BS-HL-03-4.0-12-R
- LOAD: złączce śrubowe AK500/2

**Ustawienia Fuse-bitów:**

- CKSEL3...0: 0010
- SUT1...0: 10
- CKDIV8: 0
- CKOUT: 1
- DWEN: 1
- EESAVE: 0



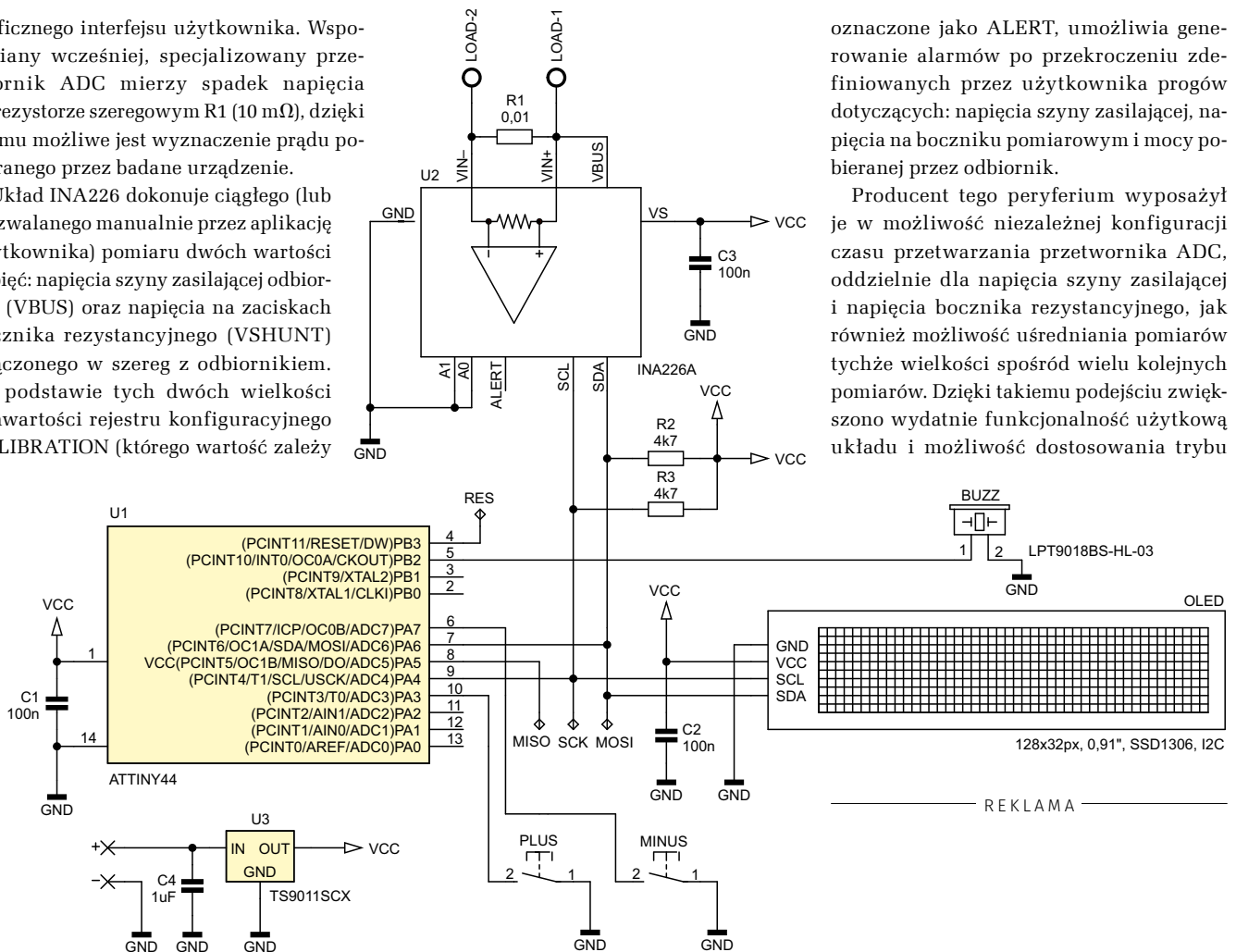
Rysunek 1. Schemat blokowy układu INA226

graficznego interfejsu użytkownika. Wspomniany wcześniej, specjalizowany przetwornik ADC mierzy spadek napięcia na rezystorze szeregowym R1 (10 mΩ), dzięki czemu możliwe jest wyznaczenie prądu pobieranego przez badane urządzenie.

Układ INA226 dokonuje ciągłego (lub wyzwalanego manualnie przez aplikację użytkownika) pomiaru dwóch wartości napięć: napięcia szyny zasilającej odbiornik (VBUS) oraz napięcia na zaciskach bocznika rezystancyjnego (VSHUNT) włączonego w szereg z odbiornikiem. Na podstawie tych dwóch wielkości i zawartości rejestru konfiguracyjnego CALIBRATION (którego wartość zależy

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.

Producent tego periferium wyposażył je w możliwość niezależnej konfiguracji czasu przetwarzania przetwornika ADC, oddzielnie dla napięcia szyny zasilającej i napięcia bocznika rezystancyjnego, jak również możliwość uśredniania pomiarów tychże wielkości spośród wielu kolejnych pomiarów. Dzięki takiemu podejściu zwiększono wydatnie funkcjonalność użytkową układu i możliwość dostosowania trybu



Rysunek 2. Schemat ideowy urządzenia Ampera

od wymaganej rozdzielczości pomiaru i parametrów zastosowanego bocznika rezystancyjnego) układ oblicza następujące wielkości elektryczne: prąd oraz moc pobieraną przez odbiornik i udostępnia je aplikacji użytkownika, ładując obliczone wielkości do stosownych rejestrów konfiguracyjnych jak również

ustawiając odpowiednie flagi zakończenia konwersji.

Ponadto, dzięki wyposażeniu go w grupę specjalnych rejestrów konfiguracyjnych odpowiedzialnych za porównywanie zmierzonych i obliczonych wartości z wartościami progowymi, jak również wyprowadzenie

jego pracy do wymagań konkretnej aplikacji. Należy jedynie pamiętać, że wydłużenie czasu przetwarzania wbudowanego przetwornika ADC wydatnie zwiększa uzyskaną dokładność pomiaru, zaś uśrednianie większej liczby próbek zdecydowanie polepsza odstęp sygnału od szumu, w związku z czym w rzeczywistych aplikacjach należy dobierać maksymalne i możliwe do zaakceptowania wartości tychże parametrów, kierując się dla przykładu szybkością zmian badanych przebiegów jako kryterium wyjściowym.

Aby poznać wszystkie możliwości drzemiące w układzie INA226, należałoby sięgnąć do jego noty aplikacyjnej lub do... mojego artykułu prezentującego urządzenie USB-tester opublikowanego na łamach naszego miesięcznika w maju 2019 roku, gdzie szczegółowo omawiam wszelkie zagadnienia implementacyjne, więc nie będę ich powielał w tym miejscu, tym bardziej że jest to dość obszerna lektura.

Kilka słów uwagi poświęcę natomiast programowej implementacji interfejsu I<sup>2</sup>C, dzięki któremu mikrokontroler współpracuje z układem INA226 (i wyświetlaczem OLED). Wykorzystałem programową implementację wspomnianego sprzęgu z dwóch powodów. Po pierwsze, mikrokontroler ATtiny44 nie został wyposażony przez producenta w pełnoprawny kontroler interfejsu I<sup>2</sup>C (w tym wypadku TWI – *Two Wire Interface*), lecz tylko w jego uproszczoną wersję o nazwie USI (*Uniwersal Serial Interface*), która nie zapewnia całkowicie sprzętowego wsparcia medium transmisyjnego, przez co w moim przekonaniu programowa

implementacja jest w pełni uzasadniona i na dodatek prostsza. Po drugie, w przypadku programowego wsparcia interfejsu I<sup>2</sup>C do wyboru mamy dowolne piny mikrokontrolera, co ułatwia zaprojektowanie obwodu drukowanego urządzenia. Z tych właśnie powodów zdecydowałem się zaimplementować sprzęg I<sup>2</sup>C na drodze czysto programowej. Inną sprawą jest fakt, że tego typu implementacja jest bardzo prosta, gdyż wspomniane medium transmisyjne jest dobrze udokumentowane i proste w swoich założeniach. Co istotne, nie będę w tym miejscu opisywał wszystkich, szczególnych cech standardu, gdyż łatwo znaleźć stosowne informacje w sieci, a skupię się

Listing 1. Plik nagłówkowy programowej implementacji magistrali I<sup>2</sup>C

```
#define I2C_BUS_FREQ 400000

#define SDA_PORT_REG PORTA
#define SDA_PIN_REG PINA
#define SDA_DDR_REG DDRA
#define SDA_PIN_NR PA6

#define SCL_PORT_REG PORTA
#define SCL_PIN_REG PINA
#define SCL_DDR_REG DDRA
#define SCL_PIN_NR PA4

#define SDA_LOW SDA_PORT_REG &= ~(1<<SDA_PIN_NR)
#define SDA_HIGH SDA_PORT_REG |= (1<<SDA_PIN_NR)
#define SDA_AS_INPUT SDA_DDR_REG &= ~(1<<SDA_PIN_NR)
#define SDA_AS_OUTPUT SDA_DDR_REG |= (1<<SDA_PIN_NR)
#define SDA_IS_LOW (!(SDA_PIN_REG & (1<<SDA_PIN_NR)))

#define SCL_LOW SCL_PORT_REG &= ~(1<<SCL_PIN_NR)
#define SCL_HIGH SCL_PORT_REG |= (1<<SCL_PIN_NR)
#define SCL_AS_INPUT SCL_DDR_REG &= ~(1<<SCL_PIN_NR)
#define SCL_AS_OUTPUT SCL_DDR_REG |= (1<<SCL_PIN_NR)
#define SCL_IS_LOW (!(SCL_PIN_REG & (1<<SCL_PIN_NR)))

#define CLK_BITWAIT (F_CPU/I2C_BUS_FREQ/2)

#define NACK 0
#define ACK 1
```

Listing 3. Funkcja odpowiedzialna za wygenerowanie sygnału Start magistrali I<sup>2</sup>C

```
void i2cStart(void){
    //Generujemy sygnał START: SDA z 1 na 0
    //w czasie gdy SCL=1
    SDA_LOW;
    _delay_us(CLK_BITWAIT);
    //SCL z 1 na 0 - przygotowanie do transmisji
    SCL_LOW;
    _delay_us(CLK_BITWAIT);
}
```

Listing 5. Funkcja odpowiedzialna za wysłanie bajta przy użyciu magistrali I<sup>2</sup>C

```
uint8_t i2cWriteByte(uint8_t Byte){
    uint8_t Ack;

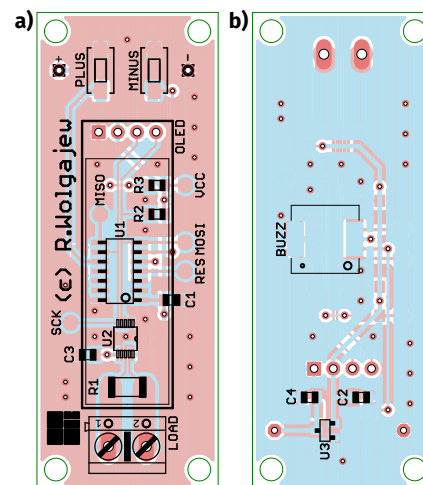
    for(uint8_t i=0; i<8; ++i){
        //SCL jest w stanie niskim
        if(Byte & 0x80) SDA_HIGH; else SDA_LOW;
        Byte <<= 1;
        //Układ Slave zatrzymuje bieżący bit
        SCL_HIGH;
        //Obsługa „clock stretching”
        SCL_AS_INPUT;
        //Czekamy, aż Slave zwolni szynę zegara
        while(SCL_IS_LOW);
        SCL_AS_OUTPUT;
        _delay_us(CLK_BITWAIT);
        SCL_LOW;
        _delay_us(CLK_BITWAIT);
    }
    //Odczytujemy 9. bit
    //potwierdzenia z układu Slave
    SDA_HIGH;
    SDA_AS_INPUT;
    SCL_HIGH;
    _delay_us(CLK_BITWAIT);
    Ack = (SDA_PIN_REG & (1<<SDA_PIN_NR))>>SDA_PIN_NR;
    SCL_LOW;
    _delay_us(CLK_BITWAIT);
    SDA_AS_OUTPUT;
    return Ack;
}
```

Listing 2 Funkcja inicjująca porty magistrali I<sup>2</sup>C

```
void i2cInit(void){
    //Porty SDA i SCL
    //jako wyjściowe ze stanem 1
    SDA_HIGH;
    SCL_HIGH;
    SDA_AS_OUTPUT;
    SCL_AS_OUTPUT;
}
```

Listing 4. Funkcja odpowiedzialna za wygenerowanie sygnału Stop magistrali I<sup>2</sup>C

```
void i2cStop(void){
    //Generujemy sygnał STOP:
    //SDA z 0 na 1, w czasie gdy SCL=1
    SDA_LOW;
    _delay_us(CLK_BITWAIT);
    SCL_HIGH;
    _delay_us(CLK_BITWAIT);
    SDA_HIGH;
    _delay_us(CLK_BITWAIT);
}
```



Rysunek 3. Schemat montażowy: a) widok strony TOP, b) widok strony BOTTOM

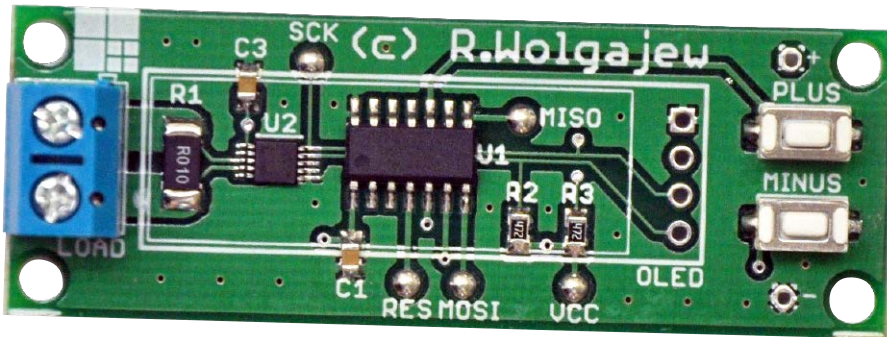
jedynie na mojej, uproszczonej, acz wystarczającej implementacji programowej.

Plik nagłówkowy implementacji definiujący wszystkie, niezbędne cechy sprzętowe pokazano na listingu 1. Na listingu 2 pokazano z kolei prostą funkcję inicjującą porty magistrali I<sup>2</sup>C, której zadaniem jest ustawienie stanów spoczynkowych. Dwie, podstawowe funkcje standardu I<sup>2</sup>C generujące sygnał Start i Stop pokazano na listingach 3 i 4. Na listingach 5 i 6 pokazano kolejne, dwie proste funkcje umożliwiające wysłanie i odebranie bajta na magistrali I<sup>2</sup>C. Funkcja wysyłająca jako rezultat swojego działania zwraca wartość bitu potwierdzenia po stronie odbiornika danych (w tym przypadku układu slave), zaś funkcja pozwalająca na odbiór bajta przyjmuje jako argument wywołania wartość bitu potwierdzenia po stronie odbiornika danych (w tym przypadku układu master).



Rysunek 4. Wygląd graficznego interfejsu użytkownika





Fotografia 1. Zmontowana płytkę od strony TOP przed przylutowaniem wyświetlacza OLED

Urządzenie wyposażono w niewielki, ale efektywny wyświetlacz OLED, którego obsługę mikrokontroler realizuje także dzięki programowej implementacji interfejsu I<sup>2</sup>C. Powodem wybrania tego typu wyświetlacza jest jego doskonała widoczność nawet w warunkach światła słonecznego i bardzo atrakcyjna cena. W tym momencie, podobnie jak poprzednio, po szczegóły implementacyjne dotyczące tegoż peryferium odsyłam ponownie do wspomnianego wcześniej artykułu, gdzie drobiazgowo opisano zagadnienia programowe związane z obsługą tego rodzaju wyświetlacza OLED.

## Montaż i uruchomienie

Schemat montażowy naszego urządzenia pokazano na rysunku 3. Zaprojektowano bardzo zgrabną, dwustronną, niewielką płytkę drukowaną, która swoim rozmiarem jest zbliżona do zastosowanego wyświetlacza OLED, udostępniając dodatkowo niezbędne otwory montażowe. Warto również podkreślić, że dla zminimalizowania rozmiaru obwodu drukowanego przewidziano tutaj montaż elementów po obu stronach laminatu. Aplikację urządzenia rozpoczynamy od przylutowania układu INA226. Najprostszym sposobem montażu elementów o tak dużym zagęszczeniu wyprowadzeń, niewymagającym jednocześnie posiadania specjalistycznego sprzętu, jest użycie zwykłej stacji lutowniczej, dobrej

Listing 6. Funkcja odpowiedzialna za odbiór bajta przy użyciu magistrali I2C

```
uint8_t i2cReadByte(uint8_t Ack){
    uint8_t Byte = 0;
    SDA_HIGH;
    SDA_AS_INPUT;

    for(uint8_t i=0; i<8; ++i) {
        Byte <<= 1;
        SCL_HIGH;
        //Obsługa „clock stretching”
        SCL_AS_INPUT;
        //Czekamy, aż slave zwolni szynę zegara
        while(SCL_IS_LOW);
        SCL_AS_OUTPUT;
        _delay_us(CLK_BITWAIT);
        if(!SDA_IS_LOW) Byte++;
        SCL_LOW;
        _delay_us(CLK_BITWAIT);
    }

    //Transmitujemy 9. bit Ack
    SDA_AS_OUTPUT;
    if(Ack) SDA_LOW; else SDA_HIGH;
    SCL_HIGH;
    _delay_us(CLK_BITWAIT);
    SCL_LOW;
    _delay_us(CLK_BITWAIT);
    SDA_HIGH;
    return Byte;
}
```

jakości cyny z odpowiednią ilością topnika oraz dość cienkiej plecionki rozlutowniczej, która umożliwi usunięcie nadmiaru cyny spomiędzy wyprowadzeń układu. Należy przy tym uważać, by nie uszkodzić termicznie tegoż elementu.

Dalej lutujemy pozostałe elementy półprzewodnikowe, potem rezystory i kondensatory oraz inne elementy bierne, a na końcu mikroprzełączniki PLUS i MINUS, buzzer piezoelektryczny i gniazdo podłączeniowe LOAD. Z uwagi na zagęszczenie wyprowadzeń układów scalonych przed pierwszym podłączeniem układu do zasilania należy

jeszcze raz sprawdzić jakość wykonanych połączeń, aby nie dopuścić do ewentualnych zwarców. Wspomniana kontrola będzie znacznie łatwiejsza, jeśli zmontowaną płytkę przemyjemy alkoholem izopropylowym w celu wypłukania nadmiaru kalafonii lutowniczej. Na samym końcu montujemy wyświetlacz OLED, lutując jego wyprowadzenia w przeznaczone do tego celu pola lutownicze (należy sprawdzić polaryzację zasilania), gdyż połączenia elektryczne zapewniają mu jednocześnie wystarczająco stabilny montaż mechaniczny. Poprawnie zmontowany układ powinien działać już po włączeniu zasilania. Na fotografii 1 pokazano zmontowane urządzenie (od strony TOP) przed przylutowaniem wyświetlacza OLED.

## Obsługa

Po włączeniu zasilania (zaciski +/- na obwodzie drukowanym) urządzenie pracuje w trybie pokazywania wartości mierzzonego prądu (rozdzielczość 1 mA), przy czym ekran w tym trybie odświeżany jest 10 razy na sekundę. Każdorazowe wciśnięcie klawiszy PLUS/MINUS powoduje chwilowe (2 s) przełączenie urządzenia w tryb pokazywania wartości alarmu dla mierzzonego prądu, co symbolizowane jest poprzez wyświetlenie symbolu dzwoneczka po lewej stronie wartości prądu. W trybie tym krótkie naciśnięcie wspomnianych klawiszy powoduje każdorazową regulację wartości prądu alarmowania o 1 mA, długie naciśnięcie o 10 mA, zaś naciśnięcie i przytrzymanie o 100 mA. Ustawiona wartość progu alarmowania zapamiętywana jest każdorazowo w nieulotnej pamięci EEPROM mikrokontrolera i wczytywana podczas włączania urządzenia. Przekroczenie ustawionej wartości alarmowania (a dokładnie jej wartości bezwzględnej) przez mierzony prąd powoduje generowanie pojedynczych dźwięków poprzez wbudowany buzzer piezoelektryczny. Wygląd graficznego interfejsu użytkownika urządzenia Ampera pokazano na rysunku 4.

Robert Wołgajew, EP