

# F-meter Modułowy licznik częstotliwości i czasu

część 1



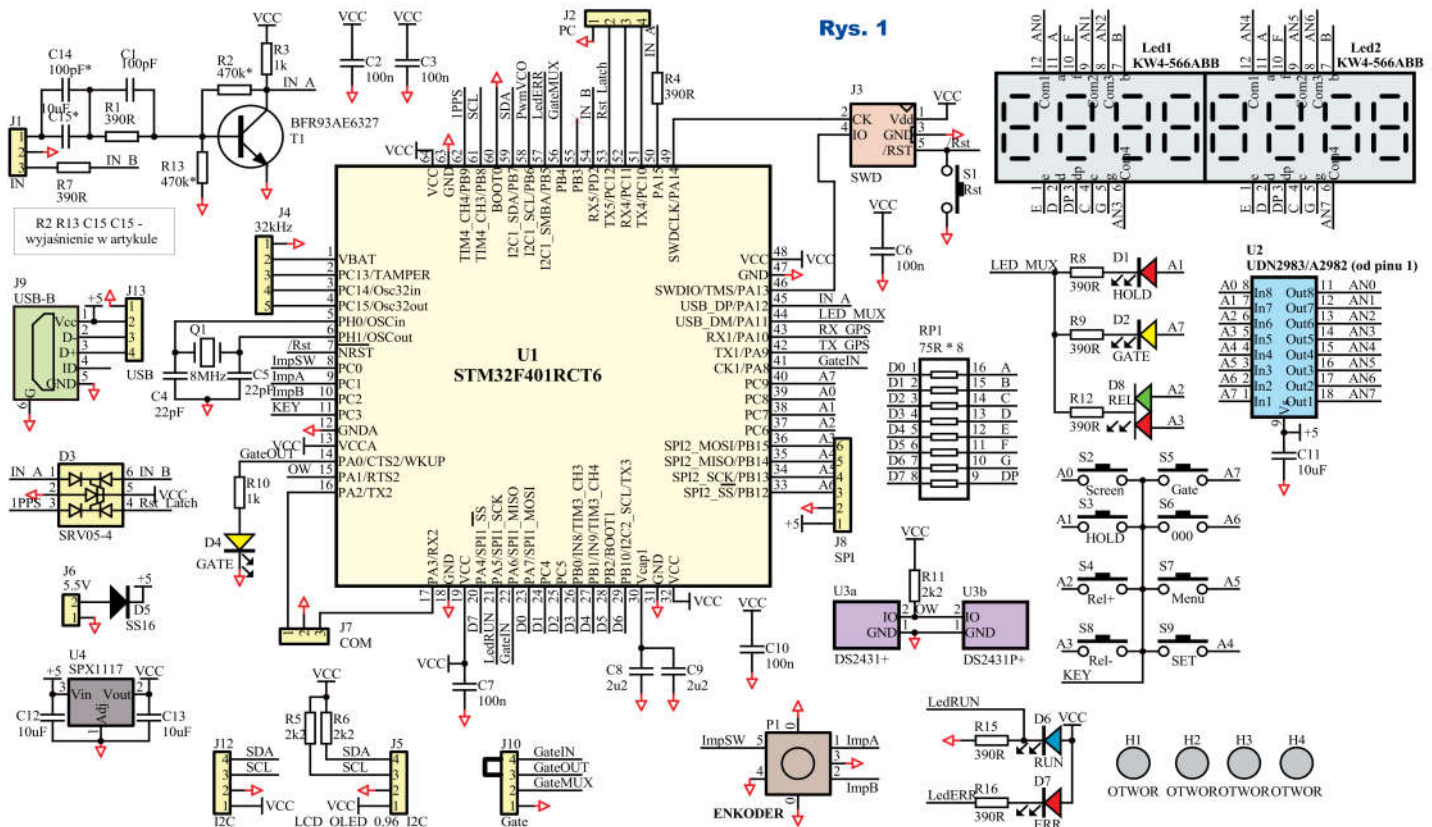
Projektów liczników częstotliwości w Internecie jest bardzo dużo, poczynając od opartych na mikrokontrolerach 8051, przez AVR, PIC do ARM. Niestety, znalezienie dobrej konstrukcji nie jest łatwe. Popularne rozwiązania oparte na PIC, jak AVT-2885, bazują na pomiarze liczby impulsów w jednostce czasu. Metoda ta ma poważną wadę przy pomiarze niskich częstotliwości – mała rozdzielczość. Aby polepszyć sytuację, trzeba rozsądnie zwiększyć czas pomiaru.

Opisywane urządzenie służy do pomiaru częstotliwości w zakresie od 0,1Hz do 42MHz i czasu do 10 sekund z rozdzielczością 23,8ns. Tym, co odróżnia prezentowany licznik od innych, jest sposób pomiaru niskich częstotliwości. Przy pomiarze częstotliwości niższych

niż 50kHz mierzony jest okres sygnału, po czym przeliczany na częstotliwość. Dzięki temu pomiar częstotliwości 1Hz z rozdzielczością 1μHz (0,000001Hz) trwa tyle co okres sygnału, czyli sekundę. Tradycyjną metodą pomiar trwałby milion sekund, czyli ponad 11,5 dnia! Poza pomiarem częstotliwości możliwy jest pomiar okresu. Można zmierzyć czas trwania stanu wysokiego i niskiego badanego sygnału. Na podstawie tych pomiarów wyliczany jest współczynnik wypełnienia z rozdzielczością 0,01%. Miernik ma dwa wejścia, drugie nie mierzy czasów, tylko częstotliwości, ale można przełączyć je w tryb licznika impulsów z przechwytywaniem i resetem. Ze względu na to, że impulsy sygnału wejściowego są zliczane w liczniku 32-bitowym, który może być taktowany częstotliwością do 42MHz, nawet przy bramkowaniu 100

sekund nie następuje przepełnienie licznika. Miernik nie ma automatycznej zmiany zakresu, tylko wynik pomiaru jest reprezentowany w inny sposób. Podobnie przy pomiarze czasu, pomiar trwający 10 sekund nie jest w stanie przepełnić licznika taktowanego 42MHz. W tym przypadku także sposób reprezentacji pomiaru jest różny dla różnych sygnałów wejściowych. Wynik pomiaru jest prezentowany na ośmiu wyświetlaczach 7-segmentowych oraz dodatkowym OLED 128x64 piksele. Mikrokontroler jest taktowany generatorem kwarcowym, ale można doprowadzić sygnał z zewnętrznego precyzyjnego generatora, który może być

Rys. 1





synchronizowany jeszcze precyzyjniejszym wzorcem, na przykład sygnałem GPS. Do licznika można podłączyć rozbudowany układ obwodów wejściowych AVT3277 sterowany magistralą I2C.

## Opis układu

Schemat ideowy pokazany jest na **rysunku 1**. Jak widać, budowa miernika nie jest skomplikowana, czego nie można powiedzieć o pierwszym, który zbudowałam ponad 30 lat temu. Miał 37 układów TTL, pobierał 1,5A, mierzył częstotliwość do 125MHz (układy rodziny 74S rodzimej produkcji) w sześciu zakresach przełączanych ręcznie przyciskami góra dół, więc był przygotowany do automatycznej zmiany zakresu, której się nigdy nie doczekał. Mierzył okres i czas trwania poziomu H, L sygnału. Wyniki były prezentowane na 6-cyfrowym wyświetlaczu 7-segmentowym. Gdy budowałam tamten miernik, nie miałam oscyloskopu. Analizatory logiczne były nieosiągalne dla przeciętnego Kowalskiego. Miałam tylko prostą sondę TTL. Z tego powodu uruchomienie miernika trwało kilka dni. Samo wiercenie PCB pochłonęło cały dzień, bo na płytce było ponad 700 otworów. Mimo że konstrukcja była dosyć nietypowa, redakcja „Radioelektronika“ nie była zainteresowana publikacją, a konkurencyjne czasopisma jeszcze nie istniały. Dlaczego nie byli zainteresowani? Tego nie wiem. Może nie chcieli uwierzyć, że młody człowiek może zaprojektować tak skomplikowane urządzenie, wykonać w domu dwuwarstwową PCB i całość działa. Teraz jest inaczej. Płytek nie trzeba robić w domu, a AVT wykona prototypowe płytki nieodpłatnie.

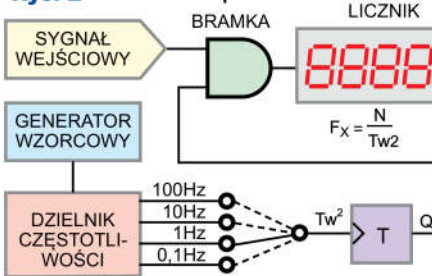
Czy teraz do uruchomienia prezentowanego licznika wymagany jest oscyloskop i/lub analizator logiczny? Tylko jeśli wystąpią jakieś problemy, to bez oscyloskopu czy analizatora stanów logicznych pewnie się nie obejdzie. Jednak szansa na bezproblemowe uruchomienie miernika są bardzo duże.

Przed opisaniem konstrukcji przybliżę metody pomiaru częstotliwości i czasu, bo czy pomiar dokonywany jest z użyciem układów TTL, CPLD/FPGA, czy mikrokontrolera, zasada pomiaru nie zmieniła się od lat. Pierwsza metoda pomiaru to metoda bezpośrednia – **rysunek 2**. Generator wzorcowy poprzez zespół dzielników wytwarza przebieg o okresie wzorcowym. W przykładzie z rysunku 2 poza podstawowym sekundowym okresem (1Hz) jest jeszcze 10 sekund (0,1Hz), 100ms (10Hz) i 10ms (100Hz). Sygnał wzorcowy trafia na dzielnik przez dwa,

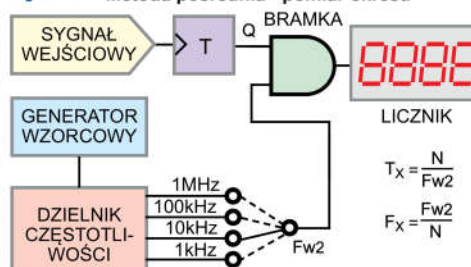
w tym przypadku zbudowany z przerzutnika T. Zakładając, że wybrano sygnał wzorcowy 1Hz, na wyjściu przerzutnika pojawi się sygnał o okresie dwóch sekund (0,5Hz), gdzie stan wysoki będzie trwał sekundę, tak samo jak i stan niski. W czasie stanu wysokiego na wyjściu bramki przepuści do licznika liczbę impulsów odpowiadającą mierzonej częstotliwości, przykładowo sygnał 1kHz spowoduje zliczenie 1000 impulsów. Taka metoda pomiaru jest dobra dla wysokich częstotliwości. Miernik cyfrowy, zależnie od momentu, w którym rozpoczyna pomiar, może zliczyć faktyczną liczbę impulsów lub o jeden mniej (zauważy jedno zbocze). W konsekwencji ostatnia cyfra może się zmieniać podczas pomiarów. Dlatego w przypadku mierników cyfrowych podaje się dokładność w procentach i cyfrach wyświetlacza. W tym przypadku będzie to  $\pm 1$ .

Gdy mierzona częstotliwość jest niska, przykładowo 10Hz, bramkując sygnałem 1 sekundy, możemy uzyskać wynik 10 lub 9. Daje to też 10% niedokładności. Dla porównania, bardzo kiepski kwarc 100ppm ma dokładność 0,01%. Wynika z tego, że dopiero gdy mierzymy częstotliwość 10kHz niedokładność kwarcu może o sobie dać znać. Z tego powodu przy pomiarze niskich częstotliwości lepsza jest metoda pośrednia – **rysunek 3**. W tym przypadku sygnał wejściowy trafia na dzielnik przez dwa, dzięki czemu na wyjściu pojawia się stan niski trwający tyle, co okres mierzonego sygnału. Generator wzorcowy generuje sygnał o wysokiej częstotliwości, który trafia na bramkę. Jak widać, zamienione zostały role generatora i sygnału wejściowego. Zakładając, że na wejście podany jest sygnał mierzony o częstotliwości 2Hz, którego okres trwa 500ms, a generator wzorcowy generuje 1000Hz (1kHz), licznik zliczy 500 impulsów. Zgodnie ze wzorem  $T_x = N / F_{wz}$ , okres sygnału mierzonego wynosi  $500 / 1000\text{Hz} = 0,5\text{Hz} = 500\text{ms}$ , częstotliwość zaś, według wzoru  $F_x = F_{wz} / N$ , wynosi  $1000\text{Hz} / 500 = 2\text{Hz}$ . Ze względu na to, że w czasach „nieprocesorowych” takie obliczenia do łatwych nie należały, liczniki częstotliwości najczęściej mierzyły metodą bezpośrednią. Liczniki na rysunkach 2 i 3 pokazano w postaci uproszczonej, nie miałyby prawa w praktyce działać poprawnie, bo nie są zerowane przed rozpoczęciem zlicza-

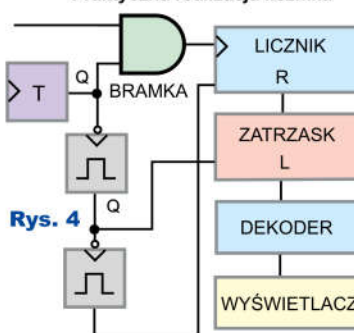
**Rys. 2** Metoda bezpośrednia



**Rys. 3** Metoda pośrednia - pomiar okresu



**Praktyczna realizacja licznika**



**Rys. 4**

nia. Praktyczna realizacja licznika wyglądać może jak na **rysunku 4**. W czasie stanu wysokiego na wyjściu przerzutnika T bramka przepuszcza impulsy, które są zliczane. Opadające zbocze sygnału na wyjściu przerzutnika wyzwala multiwibrator, który przepisuje zliczoną wartość do przerzut-

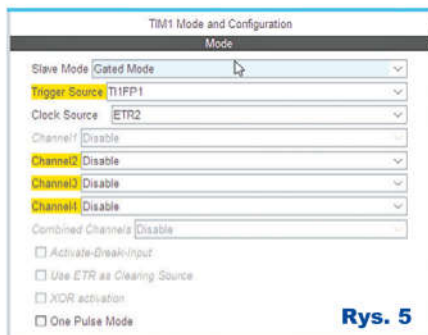
ników LATCH, natomiast opadające zbocze pierwszego multiwibratora wyzwala kolejny, który zeruje licznik, przygotowując go do kolejnego pomiaru. Bardziej rozbudowane liczniki miały synchronizację sygnału bramkującego z mierzonym przebiegiem, dzięki czemu wynik pomiaru nie zmieniał ostatniej cyfry, oczywiście o ile badany sygnał był stabilny. Czasem liczniki były budowane na bazie liczników rewersyjnych z ustawianiem. Umożliwiało to wpisanie częstotliwości pośredniej, odejmowanie jej lub dodawanie przy pomiarze. Konstrukcja takiego licznika była skomplikowana i 37 układów scalonych nie wystarczało, a i pobór prądu zdecydowanie przekraczał 1,5A.

Po (nie)zbędnej teorii przejdźmy do opisu miernika. Układ zasilany jest napięciem 5–5,5V doprowadzonym do złącza J6 lub 5V na J9. Napięcie to można nieznacznie zwiększyć. Jego górną wartość ograniczają diody LED D1, D2 i D8 sterowane z portu PA11 oraz diody wyświetlaczy LED przyłączone do portów PA i PB. Miernik można zasilic napięciem do 6,5V. Układ U4 stabilizuje napięcie dla mikrokontrolera i wyświetlacza OLED,



natomiast diody i wyświetlacze LED zasilane są z 5V. Prototyp zasilany napięciem 5,5V pobierał ok. 100mA podczas wyświetlania „0.999985”, co wydziela na stabilizatorze U4 około 500mW mocy. Do jej rozproszenia wystarczy niewielka powierzchnia miedzi na płycie. Główną rolę w mierniku odgrywa mikrokontroler U1 STM32F401RCT6, który został wybrany ze względu na niską cenę w stosunku do możliwości. W swoim wnętrzu ma 256kB pamięci FLASH, 64kB RAM, 8 timerów, w tym dwa 32-bit, po kilka UART, SPI, I2C, do tego RTC, dwa układy watchdog, wiele innych udogodnień i ma FPU, co nie jest takie oczywiste w mikrokontrolerach ARM. Mikrokontroler jest taktowany sygnałem 8MHz, ale wewnętrzny PLL powiela tę częstotliwość do 84MHz.

Mikrokontroler steruje ośmioma multipleksowanymi wyświetlaczami 7-segmentowymi. Dodatkowe informacje są wyświetlane na wyświetlaczu OLED 128×64 piksele przyłączonym do J5, który komunikuje się z mikrokontrolerem przez magistralę I2C. Linie magistrali są podciągane rezystorami 2,2kΩ. Ponad dwa razy mniejsza wartość rezystorów niż najczęściej spotykana 4,7kΩ wynika z tego, że wyświetlacz pracuje w trybie FAST (400kHz), a do magistrali mogą być podłączone kolejne płytki przyłączone do J12. Obsługę miernika umożliwia klawiatura składająca się z ośmiu przycisków oraz enkodera. Aby nie używać mikrokontrolera w większej obudowie, klawiatura obsługiwana jest, tak jak wyświetlacze – przez multipleksowanie. Ponieważ nie przewiduje się naciśnięcia więcej niż jednego przycisku w tym samym czasie, budowa klawiatury została uproszczona i nie zamontowano w niej diod separujących. To, ze względu na sterowanie jej razem z anodami wyświetlacza, powoduje fałszowanie wyświetlanej treści w przypadku naciśnięcia równocześnie więcej niż jednego przycisku oraz powoduje zwarcia pomiędzy portami mikrokontrolera. Krótkotrwałe zwarcia nie doprowadzą do ich uszkodzenia, ale takie postępowanie nie jest zalecane. Enkoder jest przyłączony bezpośrednio do wejść mikrokontrolera. Sygnał wejściowy przechodzi przez prosty wzmacniacz zbudowany na tranzystorze T1. Mimo wypróbowania dwóch rozwiązań, o czym w dalszej części artykułu, nie byłam zadowolona z ich pracy. Dlatego zdecydowałam się pominąć wzmacniacz, bo i tak planowałam zaprojektować bardziej rozbudowane obwody wejściowe. Wejścia A, B, 1PPS i LATCH/RESET



Rys. 5

mikrokontrolera zabezpiecza transil D3. Złącze J10 umożliwia doprowadzenie zewnętrznego sygnału bramkującego; jeśli ta możliwość nie jest wykorzystana, należy zewrzeć piny 3 i 4 złącza. J11 umożliwia komunikację z modulem GSM oraz sterowanie generatorem VCO. Na USART dostępny na złączu J7 wysyłane są informacje o zmierzonej częstotliwości i czasie. Parametry transmisji 921600 8N1. J8 służyła do prób z wyświetlaczem komunikującym się przez SPI. Na razie ta funkcja nie jest wykorzystana. J3 umożliwia zaprogramowanie mikrokontrolera. W J2 wykorzystane jest na razie tylko wyprowadzenie numer dwa umożliwiający przechwycenie i zresetowanie licznika B. Niewykluczone, że gdy Czytelnicy wykażą zainteresowanie poparte listami do redakcji, funkcja zatraskiwania i resetowania zostanie rozdzielona. Wtedy to wykorzystane będzie kolejne wyprowadzenie złącza J2. J4 umożliwia podłączenie rezonatora kwarcowego 32kHz i baterii, co pozwoli uruchomić RTC. Do J13 można podłączyć konwerter USB-UART, który może zapewnić komunikację z mikrokontrolerem (złącze J7). Pamięć EEPROM U3a lub U3b jest konieczna tylko wtedy, gdy wymagane jest, aby ustawienia miernika były zapamiętane po wyłączeniu zasilania. Czytelników może zastanowić, dlaczego EEPROM nie została emulowana w pamięci FLASH. Otóż STM32F401 ma pięć stron pamięci o rozmiarach 4×16kB, 64kB i 128kB. EEPROM najlepiej emulować na ostatnich stronach pamięci, w konsekwencji 192kB zostałyby stracone. Jeśli skorzystać z pierwszych stron po 16kB, aby zapisać kilkadziesiąt bajtów w emulowanym EEPROM, trzeba poświęcić 48kB FLASH (strona 0 skok do programu, strony 1 i 2 na EEPROM), tyle ile RAM miał ZX-Spectrum ©. Pod tym względem seria F4 nie jest zbyt atrakcyjna. Miałam dylemat, czy zastosować pamięć w obudowie TO-92, czy prawie dwa razy tańszą w SOIC-6. W prototypie zdecydowa-

łam się na TO-92, ale w docelowej płycie pozwoliłam wybrać użytkownikowi typ pamięci pomiędzy tanią, ale trudniejszą w montażu SOIC-6, a droższą, ale łatwą do wlutowania TO-92.

## Kilka słów o oprogramowaniu

Program napisano z wykorzystaniem CubeIDE. Zajmuje 85kB FLASH z dostępnych 256, używa 10kB RAM z dostępnych 64kB. Z dostępnych ośmiu timerów wykorzystano siedem, plus systemowy generujący przerwania co 1ms, przy czym do samego pomiaru częstotliwości i czasu wystarczą cztery.

Pomiar częstotliwości realizuje timer1 ustawiony w tryb GATED MODE – rysunek 5. **Uwaga! Ze względu na obecność mnóstwa rysunków i fotografii, nie mogą mieć wysokiej jakości w wersji drukowanej – oryginały o wysokiej jakości są dostępne w Elportalu wśród materiałów dodatkowych do tego numeru EdW.** Sygnał taktujący licznik jest pobierany z wejścia „ETR1” (PA12, pin 45), zliczanie bramkuje „TI1FP1” (PA8, pin 41). Ze względu na to, że licznik jest 16-bitowy, starsze słowo zliczane jest programowo w przerwaniu.

```
uint16_t volatile timH;
void irqTim1update(){
    timH++;
}
```

Sygnał bramkujący wywołuje przerwanie

```
uint32_t volatile Freq;
void irqTim1trigger(void){
    Freq = TIM1->CNT | timH << 16;
    TIM1->CNT = timH = 0;
    FreqAend = true;
}
```

Aby uniknąć czasochłonnych obliczeń zmiennoprzecinkowych (przeliczenie częstotliwości na czas), w przerwaniu jest wykonywana tylko prosta operacja przesuwania i dodawania oraz ustawiana flaga „FreqAend” dla pętli głównej programu. Dzięki temu czasochłonne operacje wykonywane są w pętli głównej, a nie w przerwaniu. Chwila odczytu licznika timera 1 nie ma większego znaczenia, ponieważ ze względu na bramkowanie nie zmienia on już swej wartości.

Pomiar czasu jest zrealizowany jeszcze prościej, a to za sprawą 32-bitowego licznika w trybie COMBINED MODE INPUT PWM – rysunek 6. Ze względu na to, że HAL dla STM32 nie zawsze jest wygodny, konfigurację licznika zrealizowano, pisząc bezpośrednio do rejestrów.

```
TIM2->ARR = UINT32_MAX;
TIM2->PSC = 0;
TIM2->CCMR1 = TIM_CCMR1_CC1S_0 | TIM_CCMR1_CC2S_1;
TIM2->CCER = TIM_CCER_CC2P | TIM_CCER_CC1E | TIM_CCER_CC2E;
TIM2->SMCR = TIM_SMCR_TS_2 | TIM_SMCR_TS_0 | TIM_SMCR_SMS_2;
TIM2->CR1 = TIM_CR1_CEN;
```



## Wykaz elementów

R1	10kΩ
R2	360Ω
R3	360Ω
R4	470kΩ
R5	470kΩ
R6	330R/1W
R7	470kΩ
R8	470kΩ
C1	470nF/250VAC
C2	1000uF/16V
C3	100nF
D1	1N4007
DZ1	5V1
TR1	BT136-600E
TR2	BT136-600E
U1	ATTINY13
CON1	Złącze ARK2
CON2	Złącze ARK2
CON3	Złącze ARK2
CON4	Goldpin 2×3
Podstawka 8dip	

czasami impuls wysokiego napięcia o stromych zboczach. Dodanie elementów tworzących gasik rozwiązuje problem, niestety nie ma nań miejsca na płytce.

Ostatecznie w nowej wersji pojawia się triak BT136-600E z prądem bramki 10mA. Nie jest tak bardzo czuły na krótkie przepię-

cia, co wykazały testy. Rezystory bramek są zmienione z oporności 100Ω na 360Ω.

Ponieważ wydajność prądowa istniejącego zasilacza wynosi około 20mA, liczy się każdy miliamper. Konieczna była dalej idąca oszczędność prądu i w tym celu zmniejszyłem częstotliwość taktowania mikrokontrolera do 128kHz.

Ponadto gruntownej zmianie uległ program umieszczony w mikrokontrolerze. Między innymi zmieniła się struktura programu; został napisany jako maszyna stanów. Pojawiły się również fragmenty chroniące przed zakłóceniami pochodzącymi od przełącznika.

Program napisany w języku C został skompilowany za pomocą kompilatora XC8 w wersji v2.10.

Źródło oraz wsad do procesora dostępne są w materiałach umieszczonych na Elportalu.

## Montaż i uruchomienie

Jeżeli mamy do czynienia z układem jeszcze niezmontowanym, robimy to w sposób opisany w dokumentacji kitu AVT3048, używając przy tym nowego zestawu elementów TR1, TR2, R2, R3.

Co innego, gdy przeprowadzamy modyfikację układu zmontowanego już wcześniej. Wszelkich zmian dokonujemy w ukła-

dzie całkowicie odłączonym od instalacji elektrycznej.

Na początku demontujemy triaki, a później wskazane wcześniej rezystory. Dokonujemy tego, najpierw usuwając spoiwo z punktów lutowniczych wymienianych elementów: R2, R3 i TR1, TR2, z użyciem odsysacza cyny lub miedzianej plecionki i topnika. Po wyjęciu elementów z otworów montażowych oczyszczamy pola lutownicze z topnika, używając do tego ostrego narzędzia i rozpuszczalnika w postaci IPA. Następnie montujemy nowe elementy, pamiętając o prawidłowym ułożeniu triaków, zgodnie z fotografią.

Zaprogramowanie mikrokontrolera można przeprowadzić w układzie, zasilając z programatora (dioda prostownika blokuje przepływ prądu przez diodę Zenera). Przed uruchomieniem należy sprawdzić, czy nie ma zwarcia, szczególnie w obszarze końcówek triaków.

Układ nie wymaga specjalnego uruchamiania i powinien zadziałać od razu.

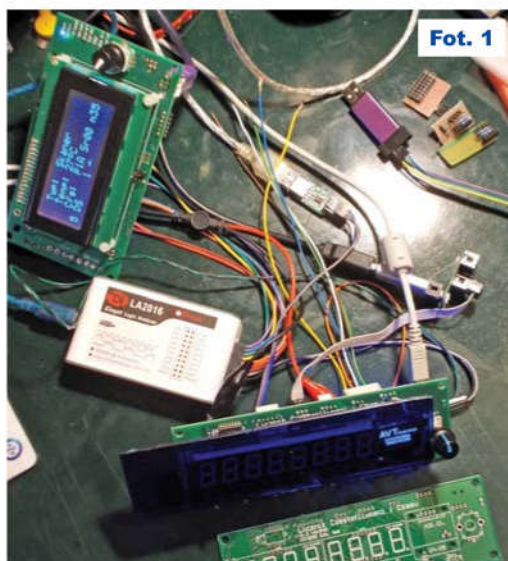
Montażu układu w instalacji powinny dokonywać osoby pełnoletnie, w pełni świadome zagrożenia ze strony prądu elektrycznego.

**Piotr Wójtowicz**

piotr.wojtowicz@elportal.pl

Ciąg dalszy ze strony 18

Podczas przenoszenia funkcji obsługi 1-Wire wspomagano się analizatorem logicznym i emulatorem/skanerem 1-Wire – **fotografia 1** (artykuł opisujący emulator znajduje się w „Elektronice Praktycznej“ 7/2020), który wraz



z programem terminala i debugerem ułatwił pracę – **rysunek 13**. Po przeniesieniu podstawowej komunikacji zrealizowano odczyt w szybkim trybie overdrive – **rysunek 14**. Po ewentualnej obsłudze EEPROM sprawdzana jest klawiatura, a właściwie bufor klawiatury obsługiwany w przerwaniach. Po tym następuje obsługa wyświetlacza OLED oraz wypełniany jest bufor wyświetlacza LED, obsługiwanego, jak się łatwo domyślić, w przerwaniu

SysTick wywoływanym co milisekundę. Na końcu pętli głównej następuje wysłanie danych na terminal.

Typowy czas obiegu pętli głównej to 2,2µs. W tym czasie AVR taktowany zegarem 16MHz wykona 35,2 taktu, co daje średnio 20 rozkazów.

Montaż i uruchomienie licznika zostaną omówione w następnym numerze

**SaS**

sas@elportal.pl

