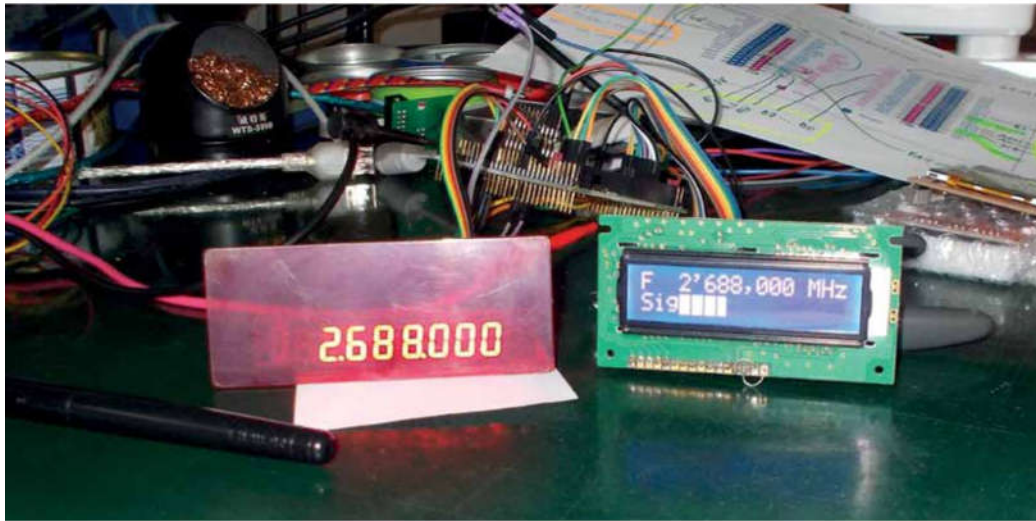


Mikroprocesorowy miernik częstotliwości

F-meter 42 MHz



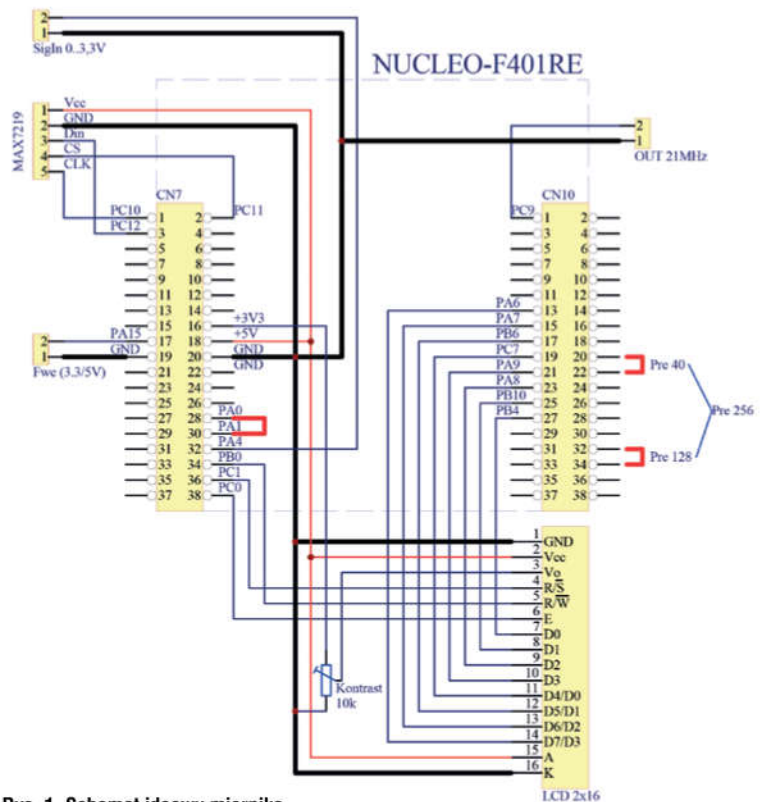
Zaprezentowany w ŚR 7-8/2020 miniaturowy F-meter miał małe wymiary, ale przez to i mały wyświetlacz. Tym razem autor przedstawia większą konstrukcję, do której można podłączyć duże wyświetlacze LED, alfanumeryczne LCD czy graficzne TFT. Ze względu na konieczność zastosowania „mocniejszego” mikrokontrolera zwiększeniu uległa także maksymalna częstotliwość, jaką licznik potrafi zliczyć i wynosi 42 MHz bez preskalera. Dodatkową funkcją jest pomiar sygnału analogowego, na przykład poziomu sygnału wejściowego.

Budowa

Aby ułatwić zmontowanie miernika, zdecydowano się na użycie płytki deweloperskiej NUCLEO-F401RE. Poza samym mikrokontrolerem zawiera także programator/debugger, dzięki czemu nie trzeba dokupować programatora. Samą płytkę programatora/debuggera można odłamać, aby zmniejszyć wymiary urządzenia. W razie konieczności zmiany oprogramowania płytki można połączyć przewodami. Zasilanie płytki NUCLEO podlega regułom znanym z ArduinoUNO/MEGA. Można ją zasilic przez złącze USB napięciem 5 V doprowadzonym do pinu +5 V Arduino lub 7-12 V doprowadzonym do Vin. W takiej sytuacji trzeba mieć na uwadze pobór prądu przez moduły wyświetlacza, bo może się okazać, że stabilizator na

płytkę NUCLEO nie jest w stanie dostarczyć wymaganej mocy. Podłączenie modułu wyświetlacza LED ze sterownikiem MAX7219 oraz wyświetlaczem LCD/OLED ze sterownikiem zgodnym z HD44780 pokazano na rysunku 1. Warto zwrócić uwagę, że potencjometr regulacji kontrastu LCD włączony jest pomiędzy masę a 3,3 V. Najczęściej potencjometr włącza się do napięcia zasilania wyświetlacza, w tym przypadku do 5 V. Takie

podłączenie jest niekorzystne, gdy miernik jest zasilany z USB komputera. Napięcie zasilania USB zależy od wielu czynników i może wahać się w granicach od 4,1 V do 5,1 V. W konsekwencji kontrast może się zmieniać zależnie od warunków zasilania. Gdy potencjometr przyłączony jest do 3,3 V, napięcie jest stabilne, bo pochodzi z wbudowanego w płytkę NUCLEO stabilizatora. Moduł z układem MAX7219 podłączony jest bezpośrednio do mikrokontrolera. Trzeba mieć świadomość, że gwarantowany poziom wysoki interpretowany przez MAX7219 przy zasilaniu 5 V wynosi 3,5 V, a STM32F401RE zasilany jest z 3,3 V. Problem ten można rozwiązać na kilka sposobów. Najlepszym rozwiązaniem jest zastosowanie konwertera poziomów choćby w postaci układu z rodziny 74HCT/ACT (nie HC/AC), wymaga to jednak dodatkowej płytki. Konwersję poziomów można by zrealizować, ustawiając wyjścia interfejsu SPI sterujące układem MAX7219 w trybie otwarty dren i podciągając je rezystorem 1 k do 5 V. Takie rozwiązanie jest możliwe, ponieważ taktowanie SPI ograni-



Rys. 1. Schemat ideowy miernika

czono do około 1,5 MHz. Koszt trzech rezystorów jest niewielki, ale gdzieś trzeba je wlutować. Innym rozwiązaniem jest obniżenie napięcia zasilania MAX7219, co realizuje dioda na module LED i wybrano tę opcję. Nie jest ona doskonała, bo spadek napięcia na diodzie będzie zależał od tego, jaki prąd nią płynie ale rozwiązanie to się sprawdza. W praktyce układ MAX7219 będzie akceptował poziom wysoki już od 3 V i konwersję można by sobie darować, ale trzeba mieć świadomość, że może to powodować problemy. Zagadnienie związane z MAX7219 ma duże znaczenie przy produkcji seryjnej, lecz w przypadku pojedynczych egzemplarzy urządzenia można pozwolić sobie na uproszczenia.

Sygnal mierzony należy doprowadzić do wyprowadzenia 17 złącza CN7 (PA15 STM32F401) płytki NUCLEO. Mikrokontroler operuje na logice 3,3 V, ale akceptuje napięcie do 5,5 V na wielu wejściach, w tym PA15. Sygnal analogowy, na przykład poziom sygnału, doprowadzony jest do wyprowadzenia 32 CN7. Dopuszczalny zakres napięć to 0–3,3 V. Jeżeli wejście nie jest używane, należy je podłączyć do masy. Na wyprowadzeniu 1 CN10 dostępny jest sygnał zegara systemowego 85MHz podzielony przez 4. Może posłużyć do testu miernika. Zwarcie z masą pinu 35 CN10 (PC4) wyświetli wynik przemnożony przez 128. Zwarcie można zrealizować, zakładając jumper pomiędzy piny 34 i 32, ponieważ 32 to masa analogowa. Zwarcie 22 CN7 (PB2) z masą (wyprowadzenie 20) wyświetli wynik pomnożony przez 40. Założenie obu zworek mnoży wynik pomiaru przez 256. Dzięki temu miernik może współpracować z preskalerem MC12080 w ustawieniu dzielenia przez 40, MC12079 dzielącym przez 128 lub 256 i każdym innym, o takich stopniach podziału. Gdy wynik pomiaru jest większy lub równy 1 GHz, wynik jest wyświetlany w MHz, a nie kHz. Na LCD jest to sygnalizowane zmianą wyświetlanych jednostek, na LED nie ma informacji o jednostkach, ale mierząc sygnał, wiadomo, jakiej wartości się spodziewać, tym bardziej że preskalery z reguły są samooscylicujące, w związku z czym mają jakąś częstotliwość minimalną. Przykładowo MC12080 wymaga minimalnej częstotliwości 100 MHz (według noty producenta, w praktyce działa od 80 MHz), MC12079 minimum 250 MHz.

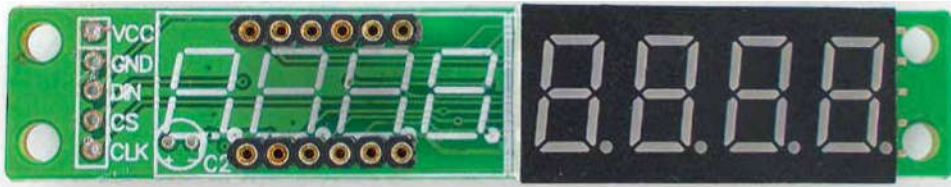
Listing 1

```
while (1){
    HAL_IWDG_Refresh(&hiwdg);
    __WFI();
    if( ! timLed ){
        timLed = 500;
        HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin );
    }
    if( ! timMax7219 ){
        timMax7219 = 500;
        FreqEnd = 0;
        MAX7219_Clr();
        uint64_t f = freq;
        if( f >= 100000000 ) { // Jesli >= 100MHz
            printU32LED( freq / 1000, ZERO_4OFF );
            if( f >= 1000000000 ) scrMAX7219[ 0 ][ 6 ] |= 128; // Kropka na pozycji GHz
        }
        else{
            printU32LED( freq, ZERO_4OFF );
            if( f >= 1000000 ) scrMAX7219[ 0 ][ 6 ] |= 128; // Kropka na pozycji MHz
        }
        scrMAX7219[ 0 ][ 3 ] |= 128; // Kropka na pozycji kHz(MHz)
        MAX7219_send_scr();
        //----- LCD
        HD44780_Home();
        char static txt[30];
        if( f > 1000000000 ) { // Jesli GHz
            f /= 1000;
            sprintf( txt, "%04d'%'03d,%'03d MHz", (int)(f/1000000), (int)((f /
1000)%1000), (int)(f % 1000) );
        }
        else{
            sprintf( txt, "%04d'%'03d,%'03d kHz", (int)(f/1000000), (int)((f /
1000)%1000), (int)(f % 1000) );
        }
        // wygaszanie pierwszych 7 zer
        for(uint8_t x=0; x<6; x++){
            if( txt[x] == ',0'){
                txt[x] = ', ';
            }
            else if( txt[x] != '0x27 ) break; // ignorujemy apostrof
        }
        txt[0] = ',F';
        if( freq < 1000000 ) txt[4] = ', '; // Gdy mniej niz 1MHz kasujemy apostrof
        HD44780_PrintString( txt );
        HD44780_SetCursorPosition( 0, 1 );

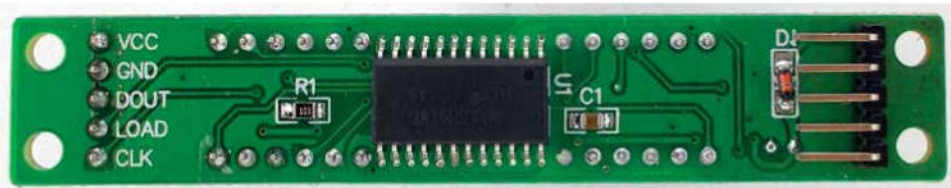
        //----- wskaźnik poziomemu sygnału
        strcpy( txt, "Sig          " );
        for(uint8_t x=0; x<13; x++){
            if( poziom >= (x+1)*6 ) txt[x+3] = 255;
            else {
                switch( poziom % 6 ) {
                    case 0: txt[x+3] = ', ';break;
                    case 1: txt[x+3] = 8; break; // znak 8 =
znak 0 a nie jest to znak końca tekstu
                    case 2: txt[x+3] = 1; break;
                    case 3: txt[x+3] = 2; break;
                    case 4: txt[x+3] = 3; break;
                    case 5: txt[x+3] = 255; break;
                }
                break;
            }
        }
        HD44780_PrintString( txt );
    }
}
```



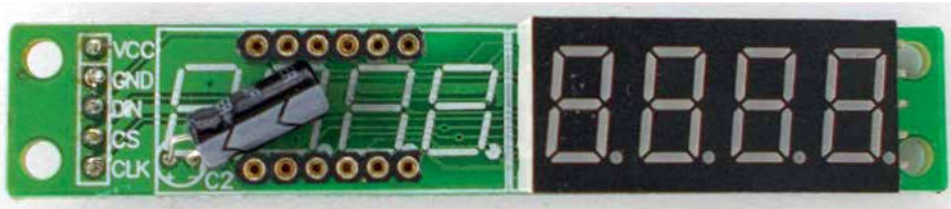
Fot. 1.



Fot. 2.



Fot. 3.



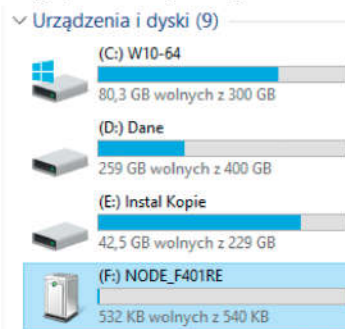
Fot. 4.

Montaż i uruchomienie

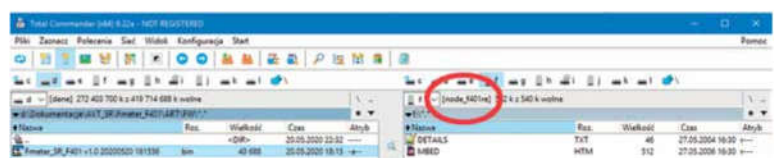
Płytkę NUCLEO łączymy z wyświetlaczem przy użyciu przewodów z wtykami żeńskimi. W przypadku wyświetlacza alfanumerycznego należy połączyć ramkę wyświetlacza z masą układu. Do tego celu na płytce wyświetlacza przewidziano miejsce na rezystor 0R lub pole lutownicze (tak zwany sierżant). Bez tego połączenia wyświetlacz może się zawieszać po przekazaniu ładunku ESD, po dotknięciu ramki wyświetlacza.

W przypadku wykorzystania modułu wyświetlacza LED zaleca się jego niewielką modyfikację. Chińskie moduły niestety zwykle są niedopracowane. Pierwszy problem to złącze goldpin (jak tanio, to tanio, goldpin jest tani i za to można włożyć wtyk niepoprawnie) zamontowany jest od strony elementów. Takie usytuowanie dobre jest do prób modułu, ale nie do jego użytkowania. Po wylutowaniu złącze można wylutować po przeciwnej stronie PCB. W prototypie, aby zmniejszyć grubość modułu, użyto kątownego goldpinu, ale lepiej umieścić tam inne, na przykład SN25-W5P. Na module znajduje się dioda DI (fot. 1) i miejsce na kondensator widoczne po wyjęciu wyświetlacza z pod-

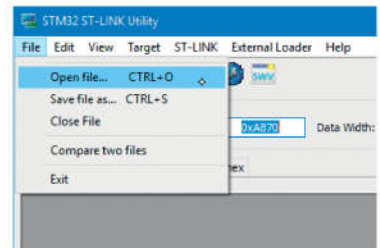
stawki (fot. 2). Zaleca się wlotowanie tam kondensatora 10–100 μ F. Wyświetlacz po przeróbce przedstawiają **fotografie 3 i 4**. Chińczycy zaoszczędzili nie tylko na kondensatorze i złączach, ale i na diodzie, która wygląda na diodę o prądzie maksymalnym 100 mA, a prąd wyświetlacza może być większy, ponieważ zmierzony w prototypie przekraczał nieznacznie 100 mA. Zaleca się usunięcie tej diody albo wlotowanie innej o większym prądzie, co może być kłopotliwe ze względu na wymiary obudowy



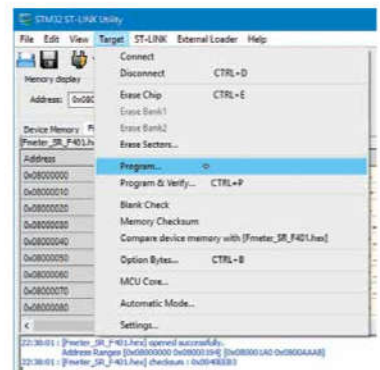
Rys. 2.



Rys. 3.



Rys. 4a.



Rys. 4b.

takiej diody. Pewnym wyjście z sytuacji byłoby zmniejszenie intensywności świecenia diod, ale ich jasność nie poraża.

Po podłączeniu wyświetlacza należy wgrać program do mikrokontrolera. Dzięki temu, że wszystkie płytki NUCLEO i DISCOVERY od STM mają wbudowany programator/debugger, czynność ta jest bardzo prosta. Program można pobrać klikając w link http://sa-s.prv.pl/fw/Fmeter_SWV_F401_v1.2_20200527_113135.ZIP. Można go wgrać na dwa sposoby. Pierwszy to przeciągnięcie ikony programu z rozszerzeniem BIN do urządzenia „NODE_F401RE” (rys. 2), drugi – skopiowanie pliku na przykład z poziomu menadżera (rys. 3).

Inną opcją jest użycie bezpłatnego programu ST-LINK Utility dostępnego na stronie producenta mikrokontrolera <https://www.st.com/en/development-tools/stsw-link004.html>. Po wczytaniu programu menu „file/Open file...” (rys. 4a) zaleca się wybrać pliki HEX lub SREC (które w przeciwieństwie do BIN są zaopatrzone w sumę kontrolną), a następnie należy zapisać jego zawartość w mikrokontroler menu „Target/Program...” (rys. 4b).

Po wgraniu należy wykonać restart mikrokontrolera przyciskiem RESET na płytce NUCLEO, o ile

nie został on wykonany przez ST-LINK Utility.

Pracę urządzenia można sprawdzić, łącząc pin 1 CN10 (PC9 – wyjście sygnału 21 MHz) z pinem 17 CN7. Na wyświetlaczu powinien pokazać się wynik pomiaru (fot. 5). Taki sygnał testowy, czy to generator/oscylatora, czy pomnożonego przez PLL lub podzielonego, aby spełnił wymagania związane z pomiarem testowym, zawsze warto sobie zapewnić. Dzięki temu można zweryfikować, czy funkcje pomiarowe działają poprawnie, bo łatwo pomylić się i wygenerować zły sygnał bramkujący zliczanie. Gdy błąd jest duży, łatwo go zauważyć mierząc sygnał wzorcowy innego generatora, ale niedoliczenie kilku impulsów (przy na przykład programowym odczycie licznika w przerwaniu) można potraktować jako rozrzut częstotliwości generatorów. W prezentowanym mierniku pomiar realizowany jest sprzętowo (tryb GATED MODE 32-bitowego licznika T2), więc błąd mógłby wynikać ze złego ustawienia licznika bramkującego T5. Dzięki temu, że sygnał bramkujący jest wyprowadzony na złącze CN7, można zdjąć zworę z pinów 28–30 i do pinu 30 (PA1) doprowadzić precyzyjny sygnał bramkujący 1000 ms. Zazwyczaj sygnał taki ma wypełnienie 50%, co oznacza, że wynik pomiaru aktualizowany jest co dwie sekundy. Sygnał generowany na PA0 (pin 28 CN7) ma wypełnienie około 1%, co oznacza, że nowy wynik pomiaru pojawia się w praktyce co sekundę. W takiej sytuacji, aby zapewnić dużą pre-

cyzję pomiaru, do wejścia OSC_IN (PH0 – pin 5 mikrokontrolera), należy doprowadzić wzorcowy sygnał 8 MHz (amplituda 3,3 V) na przykład z TCXO AVT3276. W tym celu należy wylutować rezystor 0R SB50 a sygnał doprowadzić do wyprowadzenia przeznaczonego na oscylator kwarcowy.

Na koniec można sprawdzić działanie przetwornika ADC podając na wejście PA4 (pin 32 CN7) napięcie z zakresu 0–3,3 V. Na wyświetlaczu powinien pokazać się bargraf widoczny na fotografii 6.

Jak działa program?

Wielu Czytelników zastanawia się, jak zrealizowano równoczesne wysyłanie danych do kilku wyświetlaczy, w tym „zasobożernego czasowo” kolorowego TFT, którego nie widać na fotografiach? To jest „magia” DMA. Dane do wyświetlaczy są wysyłane bez udziału μC , który realizuje inne zadania choć nie ma ich wiele, bo pomiar częstotliwości realizują sprzętowo timery. Między innymi z powodu DMA zaawansowanych timerów, dużej ilości RAM, sięgnięto w projekcie po ARM, a nie popularne AVR. Ponadto AVR potrafi mierzyć częstotliwości do max 10 MHz no i nie bez znaczenia jest fakt, że AVR wykorzystuje 25–30% szybkości komunikacji SPI wyświetlacza TFT, bo realny transfer dużej ilości danych z przepływnością 10 Mb/s jest czystą iluzją i w praktyce można osiągnąć 6–7 Mb/s. W AVR-GCC problemem jest także wyświetlanie dużych liczb, przez co programista zmuszony jest do stosowania programowych latek. Brak

obsługi typu double też czasem powoduje problemy, nie wspominając o tym, że nie ma w rodzinie AVR takich, które miałyby FPU. DMA było powodem rezygnacji z Arduino IDE, które potrafi obsługiwać niektóre STM32F4, ale biblioteki dla Arduino nie używają sprzętu. W konsekwencji podczas transmisji danych do wyświetlacza TFT, trwającej bardzo długo, bo arduinowe biblioteki stawiają pojedyncze piksele, zamiast wysyłać całe bloki danych, CPU nie robi nic innego (pomijam obsługę przerwań). Mimo że taki program zużywałby 100% mocy CPU nie dałoby się uzyskać więcej jak 3 klatek na sekundę. Tu akurat nie ma animacji i wydaje się, że czas transmisji danych do wyświetlacza nie ma znaczenia, ale gdy widać, jak rysowane są poszczególne znaki linia po linii nie wygląda to dobrze. Można by pokusić się o modyfikację bibliotek ale przeważnie są one napisane na tyle źle, że szybciej można napisać własne używające DMA. Ponadto Arduino IDE nie wspiera debuggera a bez niego praca nad programem jest bardzo czasochłonna.

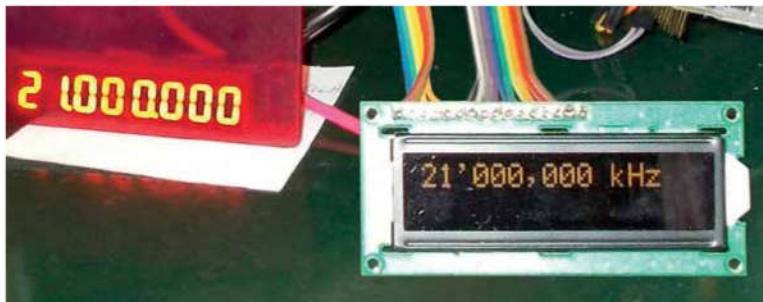
Pętla główna programu (listing 1) jest prosta, ale to nie oznacza, że reszta kodu też taka jest. Jakby tak było, to program nie zajmowałby ponad 42 kB.

Na zakończenie pierwszej części artykułu zapraszam na kanał Youtube https://www.youtube.com/channel/UCQ3i4K-fWue2P4r8PTMODyHw?view_as=subscriber, w szczególności na <https://www.youtube.com/playlist?list=PLdtkbzWTUVMlJm-f3qM6iq2tFwFflyQLFqu>, gdzie można śledzić prace nad AVT3278. Ze względu na to, że technika RF jest mi obca (wykonałem kilka tunerów FM z wykorzystaniem głowicy GFE-105), liczę na sugestie i pomoc Czytelników podczas projektowania AVT3278 oraz modyfikacji wzmacniaczy wejściowych AVT3277, aby maksymalnie wykorzystywały możliwości nowo projektowanego F-metra AVT3278.

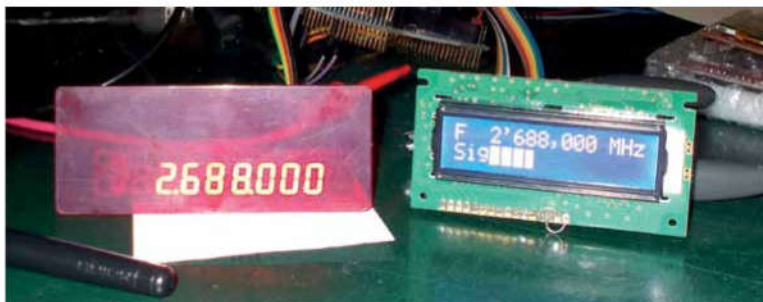
Część druga artykułu będzie poświęcona podłączeniu wyświetlacza graficznego oraz konfiguracji miernika w sposób pozwalający na uzyskanie praktycznie dowolnego mnożnika, a nie tylko 40, 128 i 256.

Najnowszych wersji programów liczników częstotliwości, dekodera Morse'a należy szukać na stronie <http://sa-s.prv.pl/fw>.

SaS
sas@elportal.pl



Fot. 5.



Fot. 6.