

Wyświetlacze VFD od podstaw, część 2



W drugiej części artykułu przedstawiamy konstrukcję przykładowego sterownika wyświetlacza VFD wykonanego na mikrokontrolerze AT89S8252.



Praktyczna realizacja układu sterowania wyświetlaczem VFD 7-LT-109 firmy Futaba

Na rys. 9 przedstawiono schemat przykładowej aplikacji układu sterowania i zasilania wyświetlacza VFD firmy Futaba typu 7-LT-109GN. Układ U1 (LM5000-3) pełni rolę przetwornicy zasilającej VFD. Ze względu na niewielkie rozmiary oraz małą liczbę elementów zewnętrznych, układ ten świetnie nadaje się do tego rodzaju aplikacji. Posiada wbudowany stopień wyjściowy mocy, którego obciążeniem może być transformator. Można również wybrać jedną z dwóch częstotliwości pracy przetwornicy, podłączając lub odłączając wyprowadzenie FS od potencjału masy. Charakterystycznym dla LM5000 jest również bardzo szeroki zakres napięć zasilających. Może on być zasilany napięciem o wartości z przedziału od 3,1 do 40 V, a maksymalna wartość przełączanego napięcia to aż 80 V. Zastosowana wersja układu umożliwi pracę przetwornicy z częstotliwością 300 lub 700 kHz. W przykładzie aplikacji przetwornica pracuje z napięciem wejściowym równym 5 V. Jest ono jednocześnie wykorzystywane do zasilania mikrokontrolera oraz driverów napięcia wyjściowego typu UDN2981. Uwaga: układ ten zasilany jest z dwóch napięć!

Obciążeniem LM5000 jest transformator wykonany przy użyciu karkasu i rdzenia pochodzących z oferty firmy Schuricht. Karkas ma numer katalogowy 334180 (karkas EFD15), a rdzeń 334160 (rdzeń do EFD15). Celowo wybrałem rdzeń, który łatwo jest kupić i który dostępny jest w stałej ofercie sprzedaży. Równolegle do uzwojenia pierwotnego transformatora dołączone są diody D1 (dioda Zenera 16 V) oraz D2 (dioda szybka np. MUR120), których celem jest ochrona stopnia wyjściowe-

go układu przed przepięciami. Pętla sprzężenia zwrotnego dołączona jest do wyjścia napięcia anodowego za diodą Schottky'ego D3 (1N5819). Wartości rezystorów R2 i R3 wpływają bezpośrednio na wartość napięcia wyjściowego. W tej aplikacji wynosi ono 22 V. Napięcie to zostało dobrane odpowiednio do typu użytego wyświetlacza.

Na karkasie nawinięto również uzwojenie podające napięcie żarzenia. Do środkowego odczepu dołączona została dioda Zenera D4 (BZX C3V9) z kondensatorem bocznikującym C12. Elementy te służą wypracowaniu napięcia odcięcia. Wartość napięcia żarzenia zmierzona oscyloskopem na doprowadzeniach żarzenia wyświetlacza wynosi około $5 V_{pp}$. Ma ono kształt zbliżony do prostokątnego. Jego wartość nie jest kontrolowana przez układ przetwornicy i nie wpływa na pracę pętli sprzężenia zwrotnego. W związku z powyższym należy unikać zwarców w obwodzie żarzenia przy pracującym układzie zasilania VFD!

Rolę układu sterowania pełni mikrokontroler AT89S8252 (U2). Pracuje on z częstotliwością taktowania równą 4 MHz. Można użyć rezonatora o innej częstotliwości, jednak trzeba się liczyć z koniecznością wykonania zmian w aplikacji sterującej. Jako układów dopasowujących poziom napięć TTL do VFD użyto UDN2981A z oferty firmy Allegro-Micro. Port P0 mikrokontrolera steruje pracą segmentów (anod), natomiast port P2 używany jest do załączania siatek znaków. Chciałbym zwrócić uwagę na fakt, że na wyjściu układu drivera U4 dołączona jest drabinka rezystorowa R4 (8 x 100 kΩ). Jest ona odpowiedzialna za wstępną polaryzację siatek znaków. Jej

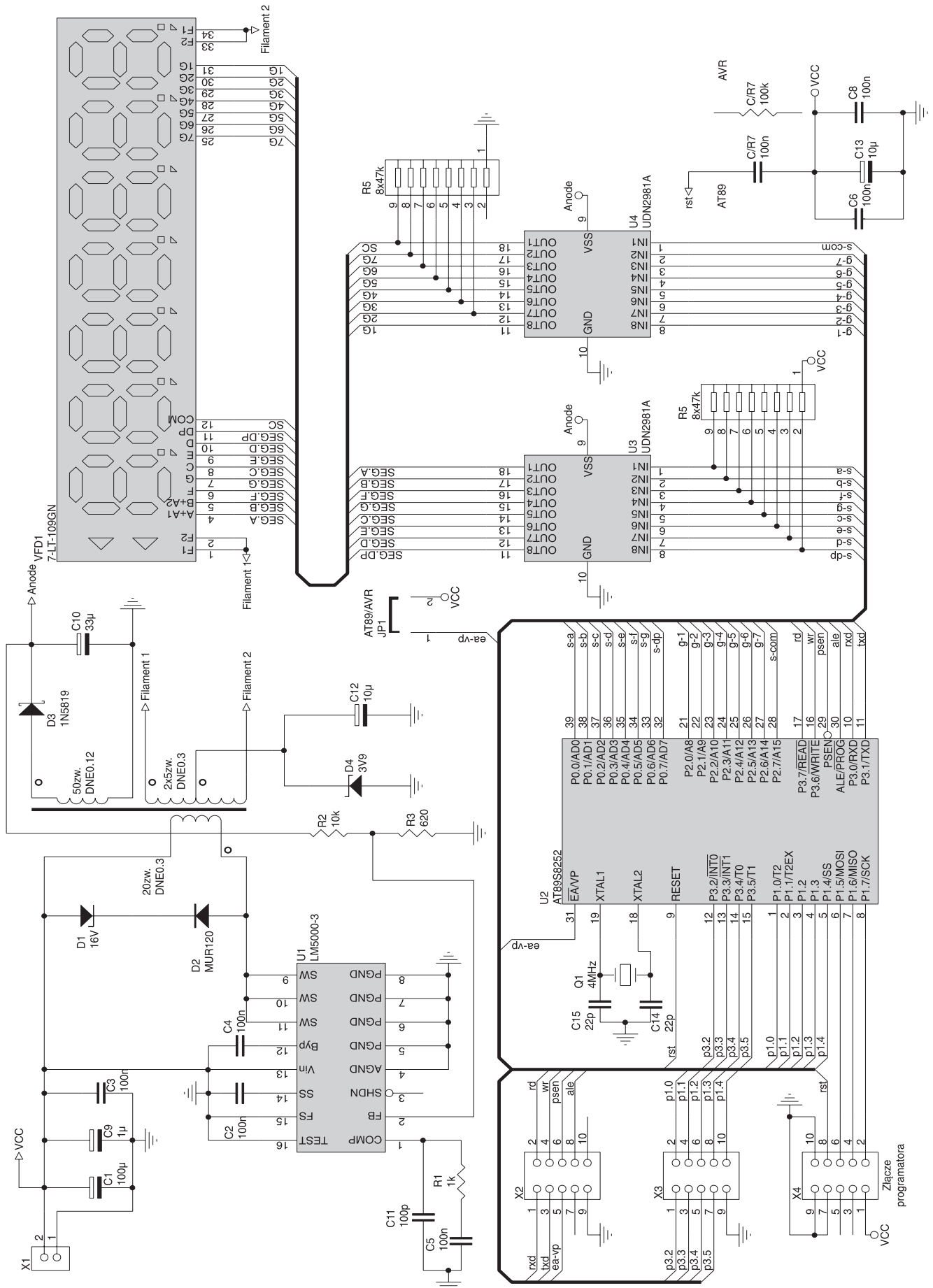
brak objawia się „widmowym” świeceniem segmentów sąsiedniego znaku.

Płytkę ewaluacyjną posiada wyprowadzone złącze do programowania mikrokontrolera w pracującym układzie (*in-circuit*) przystosowane do programatora AVT515 opisywanego w numerze 9/2003 Elektroniki Praktycznej. Można w niej stosować zarówno mikrokontroler z serii '51, jak i kompatybilny pod względem wyprowadzeń z serii AVR, na przykład AT90S8515, ATmega8535 itp. Uwaga: zmiana mikrokontrolera na AVR wymaga wlutowania rezystora w miejsce elementu C7 oraz usunięcia zwory JP1. Program sterujący napisany został w języku C dla mikrokontrolera z rdzeniem '51. Posługiwałem się wersją demonstracyjną kompilatora firmy Raisonance.

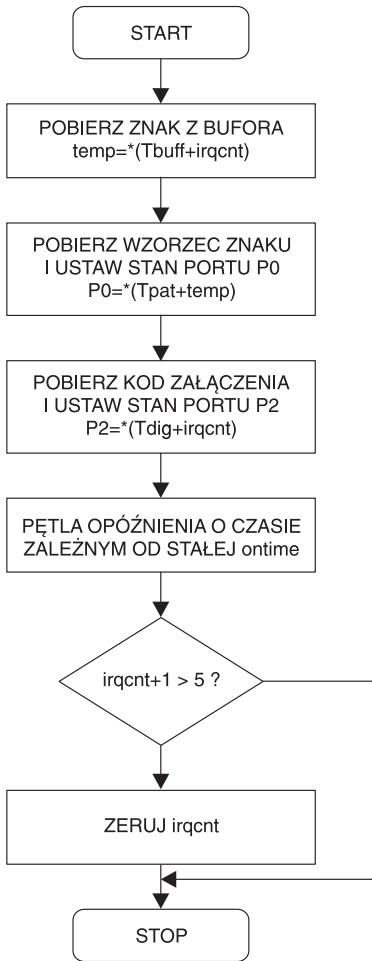
Program sterujący

Program zaczyna się od deklaracji stałych tablic mających bezpośredni wpływ na wygląd znaków oraz kolejność ich załączania. Przyjrzyjmy się tablicom, które pokazano na list. 1.

Patterns to deklaracja wzorców znaków umieszczona w obszarze pamięci programu mikrokontrolera. Umieszcza ją tam kwalifikator *code*. Typ *char* informuje kompilator o tym, że pojedynczy element tablicy ma rozmiar ośmiu bitów. Zmienna wskazująca *Tpat* pokazuje poszczególne elementy tablicy. W dalszej części programu jest ona używana jako wskaźnik początku wzorców wyświetlanych znaków, do którego dodawany jest offset w celu wyznaczenia wzorca znaku do pobrania z ROM. Stanem aktywnym, odpowiadającym za świeceniu segmentu, jest wartość logiczna „1” na pozycji bitu. Znaki w postaci cyfr uwzględniać muszą sposób dołączenia mikrokontrolera do wyświet-



Rys. 9. Schemat elektryczny sterownika wyświetlacza VFD



Rys. 10. Schemat działania funkcji obsługi przerwania

lacza VFD. Elementy tablicy ułożone są w taki sposób, że na pozycji o indeksie „0” umieszczona jest definicja znaku „0”, na pozycji „1” definicja znaku „1” i tak dalej. Ostatnim w kolejności (indeks numer 10) jest kod wygaszenia wszystkich segmentów znaku.

```

List. 1. Zmienne tablicowe zawierające kształt wyświetlanego znaku oraz
kolejność załączenia
//tablica zawierająca definicje wzorców znaków
char code patterns[11] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F,
0x6F, 0x00};
//wskaźnik do elementu tablic (wzorca znaku)
char code * Tpat = &patterns;
//tutaj kolejność załączania cyfr wyświetlacza, aktywny (załączający)
//jest stan wysoki na wyprowadzeniu portu
char code digits[6] = { 0x01, 0x02, 0x04, 0x08, 0x10, 0x20 };
//wskaźnik do tablicy kolejności załączeń
char code *Tdig = &digits;
    
```

```

List. 2. Funkcja obsługująca przerwanie Timera 1
//procedura obsługi przerwania Timera 1
//wysyłanie zmiennej 2-bajtowej do wyświetlacza - 1 znak z bufora display
void Out2Vfd(void) interrupt 3
{
    char temp;

    TH1 = irqfreq; //odświeżenie zawartości rejestru timera
    temp = *(Tbuff+irqcnt); //pobranie znaku z bufora wyświetlacza
    P0 = *(Tpat+temp); //pobranie wzorca cyfry i wyprowadzenie go przez P0
    P2 = *(Tdig+irqcnt); //załączenie cyfry

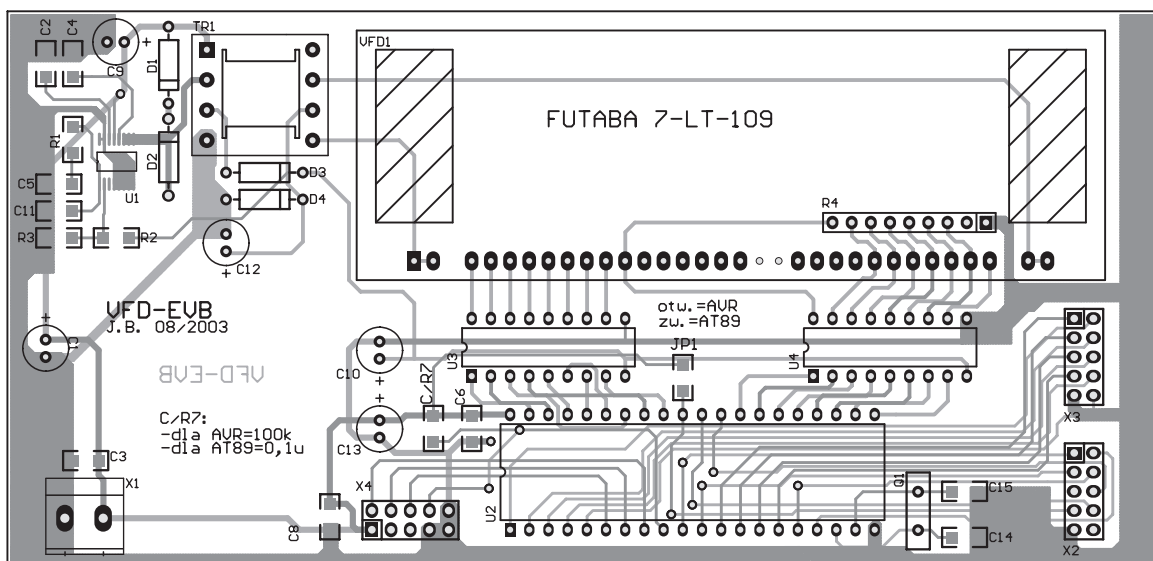
    temp = ontime;
    while (temp--); //pętla - regulacja czasu załączenia cyfry
    P2 = P0 = 0x00; //wyłączenie wszystkich cyfr (pozostałe polecenia tworzą czas Tb)

    if (++irqcnt > 5) irqcnt = 0; //jeśli to koniec cyfr do wyświetlenia
    // - powrót do początku
}
    
```

Tablica *digits* zawiera kolejność załączenia poszczególnych cyfr. W tej aplikacji znakowi o indeksie „0” odpowiada załączenie pierwszej cyfry z prawej strony wyświetlacza. Znak o indeksie 5 to pierwsza cyfra z prawej strony wyświetlacza. Kod załączenia o indeksie „7” odpowiada wyświetleniu symbolu strzałki (patrz zdjęcie wyświetlacza Futaba 7-LT-109). Łatwo jest odwrócić kolejność wyświetlania, po prostu zmieniając ustawienie elementów w tablicy *digits*. Zmienna wskazująca *Tdig* podobnie jak poprzednio, służy do wyznaczenia kodu sterującego załączeniem cyfry na bazie jej numeru.

Dynamiczne sterowanie wyświetlaniem odbywa się z wykorzystaniem funkcji obsługi przerwania Timera 1. Pracuje on w trybie ośmiobitowym z odświeżaniem zawartości rejestru timera w obsłudze przerwania. Funkcja obsługi przerwania jest bardzo „oszczędna”. Schemat jej działania przedstawiono na rys. 10.

Funkcja obsługi przerwania pobiera znak z bufora w RAM, zmienia na postać do wyświetlenia i umieszcza go na odpowiedniej pozycji wyświetlacza. Stała *irqfreq* to częstotliwość, z jaką wywoływane jest przerwanie. Częstotliwość ta jest równoważna częstotli-



Rys. 11. Schemat montażowy płytki drukowanej

List. 3. Program wykorzystany do testowania sterownika wyświetlacza

```

/*****
Obsługa wyświetlacza 6 cyfr VFD. Wyświetlanie znaków ASCII
*****/

#pragma SMALL
//dołączenie definicji rejestrów mikrokontrolera
#include <reg8252.h>

//stałe do odświeżenia zawartości timera 1
#define irqfreq0xFC
#define ontime 120
#define BLANK 10

// wzorce cyfr i liter
char code patterns[11] = {
//0 1 2 3 4 5 6 7 8 9 wygaś
0x3F, 0x06, 0x5B, 0x4F, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F, 0x00};
char code *Tpat = &patterns;
//tutaj kolejność załączania
char code digits[6] = { 0x01,0x02,0x04,0x08,0x10,0x20 };
char code *Tdig = &digits;
//bufor wyświetlacza w RAM
char data buffer[8];
char data *Tbuff = &buffer;
//licznik wejść w obsługę przerwania
char data irqcnt = 0;

//procedura obsługi przerwania od timera 1
//wysłanie zmiennej 2-bajtowej do wyświetlacza - 1 znak z bufora display
void Out2Vfd(void) interrupt 3
{
    char temp;

    TH1 = irqfreq; //odświeżenie wartości rejestru timera 1
    temp = *(Tbuff+irqcnt); //pobranie znaku z bufora wyświetlacza
    P0 = *(Tpat+temp); //pobranie wzorca cyfry i wyprowadzenie go przez P0
    P2 = *(Tdig+irqcnt); //załączenie cyfry

    temp = ontime;
    while (temp--); //pętla - regulacja czasu załączenia cyfry
    P2 = P0 = 0x00; //wyłączenie wszystkich cyfr (pozostałe polecenia tworzą czas Tb)

    if (++irqcnt > 5) irqcnt = 0; //jeśli to koniec cyfr do wyświetlenia
    // - powrót do początku
}

//program główny
void main(void)
{
    unsigned int x, D, l;

    P0 = P2 = 0; //wygaszenie cyfr, ustalenie wart.pocz. portów procesora
    for (x = 0; x<6; x++) buffer[x] = BLANK;

    TMOD = 0x10; //timer 1:16-bitowy, ustawiana tylko starsza połówka bajtu
    TH1 = irqfreq; //wartość dla reload
    ET1 = 1; //zezwozenie na przerwanie od timera 1
    TR1 = 1; //uruchomienie timera 1
    EA = 1; //zezwozenie na przyjmowanie przerwań, załączenie wyświetlania
    //wyświetlanie polega na wstawianiu cyfr do zmiennej tablicowej „buffer”
    //pozycja 0 odpowiada znakowi o najmłodszej wadze (pierwszy z prawej)
    //NA PRZYKŁAD - implementacja licznika
    x = 0;
    while (1) {
        x++;
        D = x;
        if (x > 99999) {
            l = D/100000;
            buffer[5] = l; //podstawienie do wyświetlenia:100000
            D = D - 100000*l;
        }
        if (x > 9999) {
            l = D/10000;
            buffer[4] = l; //podstawienie do wyświetlenia:10000
            D = D - 10000*l;
        }
        if (x > 999) {
            l = D/1000;
            buffer[3] = l; //podstawienie do wyświetlenia:1000
            D = D - 1000*l;
        }
        if (x > 99) {
            l = D/100;
            buffer[2] = l; //podstawienie do wyświetlenia:100
            D = D - 100*l;
        }
        if (x > 9) {
            l = D/10;
            buffer[1] = l; //podstawienie do wyświetlenia:10
            D = D - 10*l;
        }
        buffer[0] = D; //podstawienie do wyświetlenia:1
        if (x > 65534) { //czyszczenie wyświetlacza, powtórzenie cyklu
            for (x = 0; x<6; x++) buffer[x] = BLANK;
            x = 0;
        }
    }
}

```

WYKAZ ELEMENTÓW

Rezystory

- R1: 1kΩ/SMD1206
- R2: 10kΩ/SMD1206
- R3: 620Ω/SMD1206
- R4: 8x47kΩ
- R5: 8x47kΩ

Uwaga! Drabinka R5 lutowana pod płytką, nóżka 1 do wyprowadzenia 40 (VCC) mikrokontrolera.

Kondensatory

- C1: 100µF/16V
- C2...C5, C8: 100nF/SMD1206
- C/R7: 100nF/SMD 1206 dla AT89, 100kΩ/SMD 1206 dla AVR
- C9: 1µF/63V
- C10: 33µF/50V
- C11: 100pF/SMD1206
- C12, C13: 10µF/16V
- C14, C15: 22pF/SMD1206

Półprzewodniki

- U1: LM5000-3 National Semiconductor
- U2: AT89S8252-24PI zaprogramowany
- U3, U4: UDN2981A
- D1: BZX C16
- D2: MUR120
- D3: 1N5819
- D4: BZX C3V9

Różne

- VFD1: wyświetlacz FUTABA 7-LT-109GN
- Karkas do EFD15, Schuricht numer kat. 334180
- Rdzeń EFD15, Schuricht numer kat. 334160
- Dane do wykonania transformatora:
 - Uzw. pierwotne: 20 zw. DNE 0,3
 - Uzw. wtórne 1: 50 zw. DNE 0,12 (napięcie anodowe)
 - Uzw. wtórne 2: 2 x 5 zw. DNE 0,3 (napięcie żarzenia)

wości odświeżania wyświetlacza. *On-time* reguluje czas załączenia. Zmieniając tę wartość można wpływać na jasność świecenia znaku.

Program główny (list. 3) to przykład aplikacji licznika, którego stan wyświetlany jest na VFD. Licznik pracuje w pętli nieskończonej, zliczając kolejne przebiegi pętli. Po przepełnieniu jest zerowany i cykl zaczyna się od początku. Stan zmiennej licznika podawany jest konwersji na kod BCD, a poszczególne wagi dziesiętne wstawiane są do odpowiednich komórek bufora wyświetlacza. Stan bufora pobierany jest przez funkcję obsługi przerwania i wyświetlany na VFD.

Jacek Bogusz
jacek.bogusz@ep.com.pl