

# Termometr TV

Jest to dwukanałowy termometr cyfrowy, który jako wyświetlacz temperatury wykorzystuje ekran telewizora. Dzięki temu, że posiada możliwość odczytu temperatury z dwóch czujników, można dowiedzieć się, jaka temperatura panuje wewnątrz i na zewnątrz domu. Nie trzeba więc szukać miejsca na ścianie albo na regale do umieszczenia termometru. Wystarczy podłączyć termometr za pomocą kabelka do telewizora. Obraz widoczny na ekranie telewizora stanowi dwa pola, prezentujące temperaturę,

oznaczone „W” oraz „Z”, czyli temperatura wewnętrzna i zewnętrzna. Wynik pomiaru temperatury ma rozdzielczość 0,5°C. Wygląd ekranu telewizora pokazuje **fotografia 1**. Ponieważ termometr generuje obraz w formacie wideo (telewizyjnym), może współpracować tylko z telewizorami wyposażonymi w wejście wideo.

## Opis układu

Na **rysunku 1** przedstawiony jest schemat ideowy układu. Układ opiera się na mikrokontrolerze AVR ATMEGA8, który jest taktowany zewnętrznym kwarem z częstotliwością 10MHz. Czujnikami temperatury są znane układy firmy Dallas – Maxim DS18B20 (DS1820). Pracują one w konfiguracji 2-przewodowej, gdzie linia magistrali 1WIRE, czyli DQ, służy do przesyłania danych oraz zasilania układu. Jak mówi nota aplikacyjna tego układu, w konfiguracji dwuprzewodowej oprócz rezystora podciągającego linię DQ do plusa zasilania niezbędny jest tranzystor, który podaje pełne napięcie zasilania na linię DQ, gdy następuje konwersja temperatury wewnątrz układu. Układ potrzebuje wtedy

więcej mocy na dokonanie pomiaru temperatury i jej zamianę na postać cyfrową. Na schemacie tranzystor ten oznaczony jest jako T1 i jest załączany w odpowiednim momencie przez mikrokontroler U1. Wyjście układu stanowi prosty przetwornik D/A zbudowany na rezystorach R1, R2 i R3 oraz wtórnik emite-rowy na tranzystorze T2, który podaje gotowy sygnał wideo na wejście telewizora. Na wyjściu przetwornika D/A otrzymujemy jedno z trzech wartości napięcia, potrzebnych do skonstruowania ramki sygnału wideo, czyli wartości napięcia odpowiadające poziomom synchronizacji, wygaszania oraz poziomowi koloru białego. Wybór odpowiedniego napięcia polega na ustaleniu na wyjściu procesora jednego z trzech stanów: wysokiej impedancji (pin pracuje jako wejście), poziomu „0” oraz poziomu „1”. Chętnych do zapoznania się ze szczegółami budowy ramki wideo odsyłam do mojego artykułu w EdW 11/2004 na stronie 52, gdzie opisałem prostą grę TV również wykorzystującą telewizor jako wyświetlacz.

Program dla procesora napisałem w języku C, wykorzystując do tego kompilator z pakietu *WinAVR-20060421*, program obsługujący programator *STK-200 avrdude-4\_3\_0-win32* oraz środowisko programistyczne *AVR Studio v4.12 SP3*.

Kod źródłowy podzielony jest na trzy pliki:

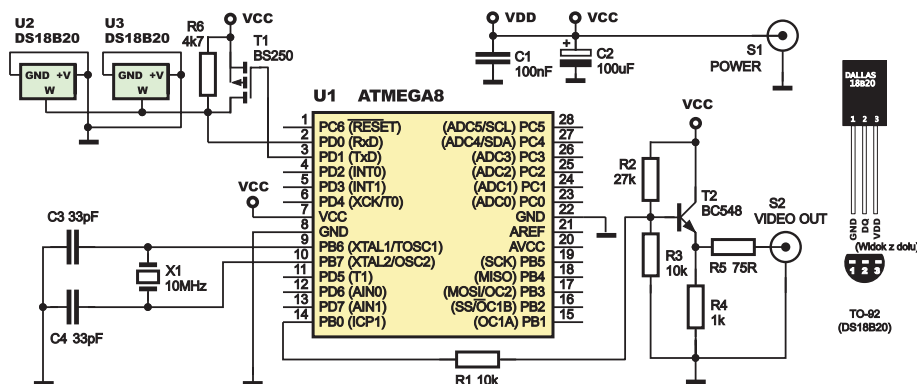
- termoTV.c – główny kod programu;
- tv.c – kod części programu odpowiedzialnego za generowanie sygnału TV i konwersję danych odczytanych bezpośrednio z układów DS. na liczbę;
- 1WIRE.c – funkcje komunikacji z czujnikami DS oparte w dużej części na dokumentacji [pdfserv.maxim-ic.com/en/an/app162.pdf](http://pdfserv.maxim-ic.com/en/an/app162.pdf)

Kod głównego programu widoczny jest na **listingu 1**. Na początku program sprawdza liczbę dostępnych na magistrali 1WIRE



Fot. 1

Rys. 1 Schemat ideowy



układów DS. Jeśli nie znajdzie żadnego albo będzie ich więcej niż 2, to wykonywanie programu kończy się, a ekran TV jest pusty. Jeśli na magistrali 1WIRE program wykryje tylko jeden układ, to wtedy temperatura odczytana z niego jest wyświetlana na wyświetlaczu „Z” i wyświetlaczu „W”. Może to być pomocne podczas uruchamiania układu i daje pewność, że oba termometry są dobrze podłączone, ewentualnie daje znak, że któryś z czujników

zawodzi. Następnie program wchodzi w nieskończoną pętlę i na przemian pobiera dane z czujników temperatury, wyświetlając je na ekranie telewizora. Niestety, odczyt z czujnika trwa nieco ponad 30ms i jest to zbyt dużo, aby wkomponować go w część ramki wideo, gdyż najdłuższy czas dostępny bez przerywania nadawania sygnału wideo wynosi 59,3µs (czas trwania poziomu wygaszania w linii wygaszania pionowego). Powoduje to przery-

wanie co kilka sekund generowania ramki wideo i w konsekwencji chwilowy zanik obrazu. Dodatkowo, żeby obraz się nie zniekształcał, na początku każdej projekcji są wyświetlane 4 czarne obrazy, które powodują jej łagodny start.

Pomiar temperatury i zapisanie wyniku do zmiennych *tempW* i *tempZ* realizuje funkcja *readDevice()*, której kod przedstawia **listing 2**. Argument tej funkcji to numer układu (liczba ze zbioru {1,2}), z którego funkcja odczyta temperaturę. Numer układu to numer indeksu, pod jakim umieszczony został jego numer seryjny w tablicy *FoundROM[nr\_ukladu][bajt\_numeru\_seryjnego]*, która została zapełniona podczas wykonywania algorytmu szukania układów na magistrali 1WIRE. Więcej informacji na temat tego algorytmu można znaleźć w dokumencie dostępnym na stronie WWW firmy Maxim: [pdfserv.maxim-ic.com/en/an/app187.pdf](http://pdfserv.maxim-ic.com/en/an/app187.pdf). Podczas projektowania układu miałem pod ręką po jednym układzie DS1820 i DS18B20. Oba czujniki różnią się od siebie między innymi sposobem zapisu temperatury w strukturze scratchpad. Ponieważ nie miałem dwóch czujników tego samego typu, postanowiłem napisać funkcję odróżniającą typ układu DS i odczytującą temperaturę z obu typów czujnika. Dzięki temu układ jest bardziej uniwersalny. W układzie 1820 rozdzielczość jest stała i wynosi 9 bajtów, a dokładniejszy pomiar jest możliwy po dokonaniu specjalnych obliczeń, włączeniu do nich wartości z 6 i 7 bajtu zawartych w strukturze scratchpad, natomiast w układzie DS18B20 mamy gotowy zapis, a rozdzielczość wybieramy, ustawiając bity w rejestrze konfiguracyjnym struktury scratchpad. Dlatego pomiar odbywa się zawsze z rozdzielczością 9-bitową, a dodatkowo bity zmiennej, do której zostaje przypisany odczyt z DS18B20, zostają przesunięte 3 miejsca w prawo, tak żeby zarówno przy odczycie z DS18B20, jak i z DS1820 funkcja zwróciła zmienną 16-bitową z takim samym formatem zapisu temperatury. Jest to konieczny zabieg, ponieważ przy 9-bitowej rozdzielczości w układzie DS18B20 na znaczeniu tracą trzy najmłodsze bity młodszego rejestru temperatury (bajt 00h struktury scratchpad). Rozróżnienie pomiędzy dwoma typami czujnika polega na sprawdzeniu czwartego bajtu struktury scratchpad (rejestr konfiguracyjny), który dla układu DS1820 ma zawsze wartość FFh, zaś ten sam rejestr w DS18B20 nigdy nie ma wartości FFh, ponieważ jego najstarszy bit ma z góry ustaloną wartość „0”.

Na początku funkcji następuje wyłączenie tranzystora pull-up zasilającego układy DS poprzez końcówkę DQ. Realizuje to funkcja *Power\_OW()*, której argumentem może być stała ON lub OFF o nazwie oznaczającej stan tranzystora jako klucza. Następnie za pomocą funkcji *matchROM()* zostaje wybrany układ, który ma zostać skalibrowany na rozdzielczość

**Listing 1** Kod funkcji *main*

```
int main() {
    unsigned char t[2][2] = { {1,1}, {1,2} };
    int tempW, tempZ;
    unsigned char temp[2];
    unsigned char count1, count2;
    unsigned char s1, s2, n;

    power_OW(OFF); //1WIRE pull-up OFF
    PORTD |= 0B00000001; //PINO pull-up
    DDRB = 0xFF; //jako wyjścia
    DDRD = 0xFF; //jako wyjścia

    n = FindDevices(); /*liczba podłączonych czujników:
                        jeśli zero lub więcej niż dwa to brak obrazu
                        jeśli jeden to wyświetlacz W i Z pokazuje tą samą temp
                        jeśli dwa to normalna praca */
    if((n>0) && (n<3)) {
        for(;;) {
            tempW = readDevice(t[n-1][0]); //odczytanie temp wewnętrznej
            TVframe(tempW, tempZ); //wyswietlenie temperatury Z i W na ekranie TV
            tempZ = readDevice(t[n-1][1]); //odczytanie temp zewnętrznej
            TVframe(tempW, tempZ); //wyswietlenie temperatury Z i W na ekranie TV
        };
    };
    return 0;
};
```

**Listing 2** Kod funkcji odczytującej temperaturę z czujników DS

```
int readDevice(unsigned char nrDev)
{
    power_OW(OFF);

    //unsigned char get[10];
    unsigned char i, who_is;
    char temp_lsb, temp_msb;
    int tempValue = 0;

    matchROM(nrDev);
    write_byte(0x4E); //konfiguracja DS18B20 na rozdzielczosc 9-bitów
    write_byte(0x00); //tl
    write_byte(0x00); //th
    write_byte(0x1F); // 00011111 BIN
    delay(94); //tconv

    matchROM(nrDev);
    write_byte(0xBE); // Read Scratch Pad
    //zapis LSB i MSB Scratch Pad'a do zmiennej zwracanej
    temp_lsb = read_byte();
    temp_msb = read_byte();

    tempValue = (int)temp_msb;
    tempValue <<= 8;
    tempValue += temp_lsb;

    read_byte();
    read_byte();
    who_is = read_byte(); //configuration byte

    matchROM(nrDev);
    write_byte(0x44); // Start Conversion - T command

    power_OW(ON);

    //test polegający na odczycie bajtu konfiguracyjnego (0x04)
    //dla DS1820 zawsze ma wartosc 0xFF
    //dla DS18B20 nigdy nie ma wartosci 0xFF

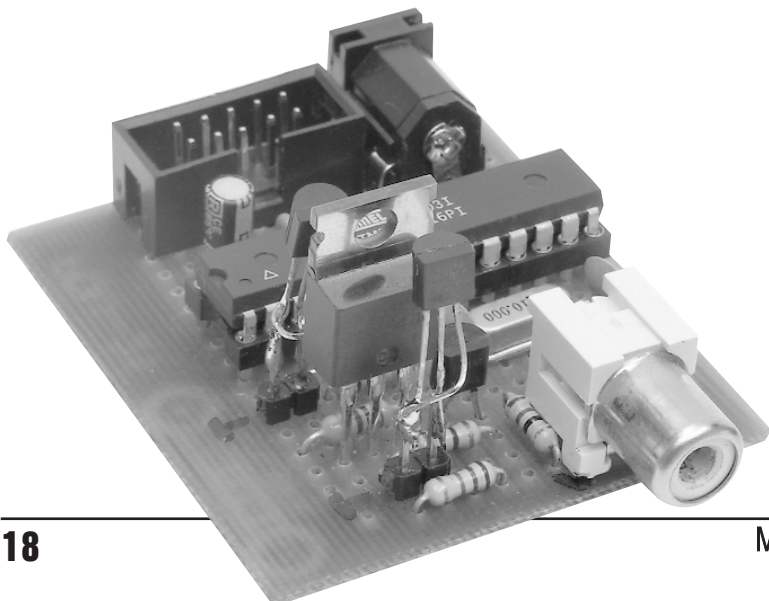
    if(who_is == 0xFF)
        return tempValue; //1820
    else
        return (tempValue >> 3);
};
```

9 bitów. Zostanie skalibrowany oczywiście tylko model DS18B20, DS1820 nie ma takiej możliwości, mimo tego nie spowoduje to jego nieprawidłowego działania, ponieważ rejestr konfiguracyjny (04h) ma status zarezerwowanego i jego wartość nie stanowi żadnej informacji dla bloku sterującego. Funkcja *match-ROM()* używa do wybierania układów polecenia 55h, a numer seryjny pobierany jest z tablicy, gdzie został zapisany w trakcie wykonywania algorytmu wyszukiwania układów na magistrali 1WIRE zaimplementowanego w funkcji *Next()*. W kolejnym kroku znów wybierany jest układ i odczytana zostaje z niego temperatura do zmiennych *temp\_lsb* i *temp\_msb*. Później następuje konwersja dwóch bajtów na 16-bitową zmienną ze znakiem. Na koniec wysłany jest rozkaz 1WIRE pomiaru temperatury do wybranego układu oraz załączony jest tranzystor pull-up. W trakcie pomiaru temperatury przez czujnik zostanie wyświetlony obraz TV.

Obraz prezentowany na ekranie telewizora jest przygotowany jako tablica 360-elementowa *matrix[]*. Wcześniej do tablicy 12-elementowej *disp[]* zostają przypisane cyfry z tablicy cyfr *znaki[]* oraz znaki dodatkowe, czyli „+”, „-”, „.”, „W”, „Z”. Każdy znak jest zapisany jako zmienna 32-bitowa, która przybiera jedną z wartości stałych zadeklarowanych w pliku TV.h. Każdy bit jest pikselem w reprezentacji tej liczby jako maczy 5x6, co lepiej widać na rysunku 2. Na obraz składa się zatem 12 takich znaków ułożonych w sześć kolumn i dwa rzędy, prezentujących temperaturę na zewnątrz i wewnątrz,

29	28	27	26	25
24	23	22	21	20
19	18	17	16	15
14	13	12	11	10
9	8	7	6	5
4	3	2	1	0

**Rys. 2 Matryca znaku z numerami bitów znaku zdefiniowanego jako stała 32-bitowa. Najstarsze 2 bity pozostają niewykorzystane. Przykładowo cyfra „2” jest zapisana jako 01111000010111010000111100000(BIN)**

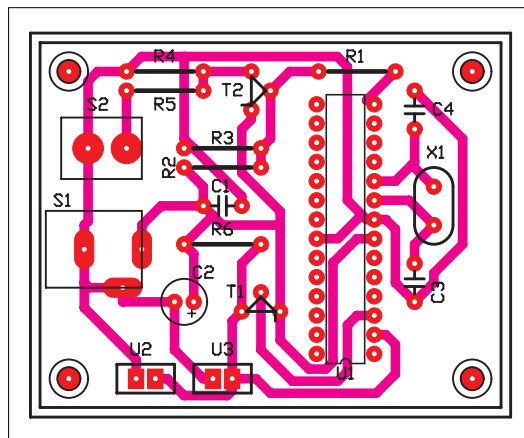


co widać na fotografii na początku artykułu.

Fragment funkcji, który zajmuje się przypisaniem odpowiedniego znaku do poszczególnych obszarów na ekranie, a następnie utworzenie gotowej tablicy *matrix[]* znajduje się na **listingu 3**. Warto zwrócić uwagę na sposób obliczania wartości bezwzględnej dla temperatury mniejszej od zera – bity zmiennej zostają zanegowane, a potem zostaje do niej dodana liczba 1.

Po przygotowaniu obrazu do wyświetlenia program wchodzi w pętlę, w której zawarty jest kod generujący ramkę wideo. Zmienna *l* jest licznikiem obiegów pętli i jej wartość początkowa decyduje, jaki czas trwa wyświetlanie obrazu między kolejnymi pomiarami. Maksymalny czas jest ograniczony długością tej zmiennej (8 bitów), czyli wynosi  $255 * 40ms = 10,2s$ , gdzie 40ms to czas wyświetlania jednego obrazu.

Kod generujący ramkę był najbardziej uciążliwym fragmentem do napisania ze względu na ściśle określone ramy czasowe wykonania każdej instrukcji, zwłaszcza, że kompilator zmieniał kod w nieprzewidywalny sposób – przy drobnej zmianie w kodzie należało na nowo mozolnie ustalać liczbę obiegów w pętlach opóźniających. Bez środowiska takiego jak AVR Studio, które umożliwi pełną kontrolę nad czasami wykonywania fragmentów kodu, byłoby to niemożliwe. Fragment kodu, który wyświetla jedną linię obrazu na ekranie, widać na **listingu 4**.



**Rys. 3 Schemat montażowy**

**Listing 3 Fragment kodu przygotowujący obraz do wyświetlenia na ekranie**

```
//tempW
disp[0] = W;
(tempW < 0) ? (disp[1] = MINUS) : (disp[1] = PLUS);
disp[2] = znaki[((tempW / 2) / 10)]; //dziesiątki
disp[3] = znaki[((tempW / 2) % 10)]; //jedności
disp[4] = KROPKA;
((tempW & 0x00000001) == 0x00000001)
? (disp[5] = PIEC) : (disp[5] = ZERO);

//tempZ
disp[6] = z;
if(tempZ < 0) {
    disp[7] = MINUS;
    tempZ = (-tempZ) + 1;
}
else {
    disp[7] = PLUS;
};
disp[8] = znaki[((tempZ / 2) / 10)]; //dziesiątki
disp[9] = znaki[((tempZ / 2) % 10)]; //jedności
disp[10] = KROPKA;
((tempZ & 0x00000001) == 0x00000001)
? (disp[11] = PIEC) : (disp[11] = ZERO);

pixel = 0;

for(d=0;d<12;d++)
    for(p = ((d/6) * 6); p < (((d/6) * 6) + 6); p++)
        for(b = 30-((d%6)*5); b > 25-((d%6)*5); b--)
            ((disp[p] >> b-1) & 0x00000001) == 0x00000001
            ? (matrix[pixel++] = 0xFF) : (matrix[pixel++] = 0x00);
```

**Listing 4 Kod wyświetlający jedną linię obrazu TV**

```
//LP 289 - 51 - 53 = 192x
//wyswietlany obraz
for(j=0;j < 192;j++) {
    setSyncOut();
    for(i=0;i<8;i++);
    setBlackOut();
    for(i=0;i<27;i++);
    asm("nop");
    asm("nop");
    if(l<4) { //4 obrazy czarne
        w celu wyeliminowania „skakania obrazu”
        asm("nop");
        for(i=0;i<79;i++);
    }
    else {
        pixel = (((unsigned int)j) / 16) * 30;
        for(i=0;i<30;i++) {
            PORTB = matrix[pixel+(unsigned int)i];
            DDRB = matrix[pixel+(unsigned int)i];
        };
    };
    setBlackOut();
    for(i=0;i<28;i++);
    asm("nop");
    asm("nop");
    if(j < (192 - 1)) {
        for(i=0;i<0;i++);
    }
};
```

## Montaż i uruchomienie

Układ montujemy na płycie drukowanej pokazanej na **rysunku 3**. Podzespoły montujemy w kolejności od części najmniejszych, kończąc na tych o największych gabarytach. Pod procesor stosujemy podstawkę. Ponieważ czujniki DS mierzące temperaturę na zewnątrz będą narażone na działanie wilgoci, należy je umieścić w szczelnej obudowie, aby uchronić ich wyprowadzenia przed korozją. Można je na przykład zalać klejem epoksydowym w cienkiej rurce metalowej wykonanej z części anteny teleskopowej radioodbiornika FM. Przed umieszczeniem czujnika w obudowie należy zlutować ze sobą końcówki GND i Vcc (nóżki 1 i 3). Czujniki z układem łączymy przewodem ekranowanym, najlepiej mającym grubą izolację, która zapewni jego wytrzymałość mechaniczną. Urządzenie wymaga zasilania napięciem 5V i jeśli istnieje możliwość wykorzystania do tego celu zasilacza antenowego, należy dodać stabilizator LM7805.

Uruchomienie termometru nie sprawia żadnych problemów, ponieważ jest to urządzenie w pełni cyfrowe, niewymagające żadnych kalibracji ani ustawień. Jedynym problemem jest ustalenie, który czujnik jest zewnętrzny, a który wewnętrzny. Można dojść do tego doświadczalnie, podgrzewając jeden z czujników i obserwując zmiany temperatury na ekranie TV.

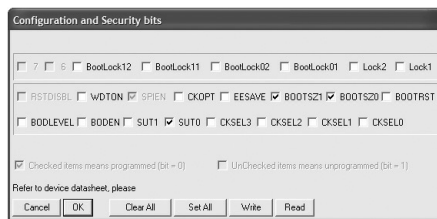
Konfigurację fuses -bitów procesora U1 jako zrzut z pracującego programu PonyProg2000 przedstawia **rysunek 4**.

Płytkę układu ma niewielkie rozmiary i łatwo będzie dopasować do niej jakąkolwiek obudowę. Na płycie przewidziane są miejsca na gniazdo zasilania oraz wyjście video, nie ma natomiast gniazd dla czujników DS. Umieszczając układ w obudowie, trzeba przemyśleć kwestię podłączenia czujników do układu - czy na stałe mają być połączone kabelkiem, czy też lepszym rozwiązaniem będzie użycie gniazd dla obu czujników.

Termometr podłączamy do telewizora za pomocą kabla video do złącza EURO lub chinch zależnie od modelu. Po włączeniu zasilania powinny pokazać się wartości +85.0 stopni, co oznacza, że nastąpił odczyt temperatury z rejestrów czujników, ale jeszcze nie nastąpił jej pomiar. Dopiero po dwóch prze-

rwach w nadawaniu obrazu od momentu włączenia napięcia zasilania (co świadczy o dokonaniu dwóch pomiarów temperatury) uzyskujemy na ekranie wartości odczytane z obu czujników.

**Piotr Wójtowicz**  
pw@elportal.pl



**Rys. 4 Konfiguracja fuses -bitów procesora U1**

### Wykaz elementów

R1	10kΩ
R2	27kΩ
R3	10kΩ
R4	1kΩ
R5	75Ω
R6	4,7kΩ
C1	100nF ceramiczny
C2	100μF
C3,C4	33pF
T1	BS250
T2	BC548
U1	ATMEGA8
U2,U3	DS18B20
X1	10MHz
S1	gniazdo zasilania 1,4/3,4 druk
S2	gniazdo chinch druk
Podstawka DIP-28	

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2817.