

Dział „Projekty Czytelników” zawiera opisy projektów nadesłanych do redakcji EP przez Czytelników. Redakcja nie bierze odpowiedzialności za prawidłowe działanie opisywanych układów, gdyż nie testujemy ich laboratoryjnie, chociaż sprawdzamy poprawność konstrukcji.

Prosimy o nadsyłanie własnych projektów z modelami (do zwrotu). Do artykułu należy dołączyć podpisane **oświadczenie, że artykuł jest własnym opracowaniem autora i nie był dotychczas nigdzie publikowany.** Honorarium za publikację w tym dziale wynosi 250,- zł (brutto) za 1 stronę w EP. Przesyłanych tekstów nie zwracamy. Redakcja zastrzega sobie prawo do dokonywania skrótów.

Pokojowy regulator temperatury ze zdalnym sterowaniem, część 1



Pomysł budowy urządzenia powstał, gdy ze względu na niską temperaturę panującą w pokoju, autor zmuszony był do korzystania z dodatkowego ogrzewania elektrycznego. Użytkowanie typowego ogrzewacza wewnętrznego (popularnej „farelki”), oprócz wysokich kosztów zużytej energii elektrycznej, miało poważną wadę: każdorazowe włączenie i wyłączenie urządzenia wymagało interwencji użytkownika. A przecież jesteśmy tak leniwi...



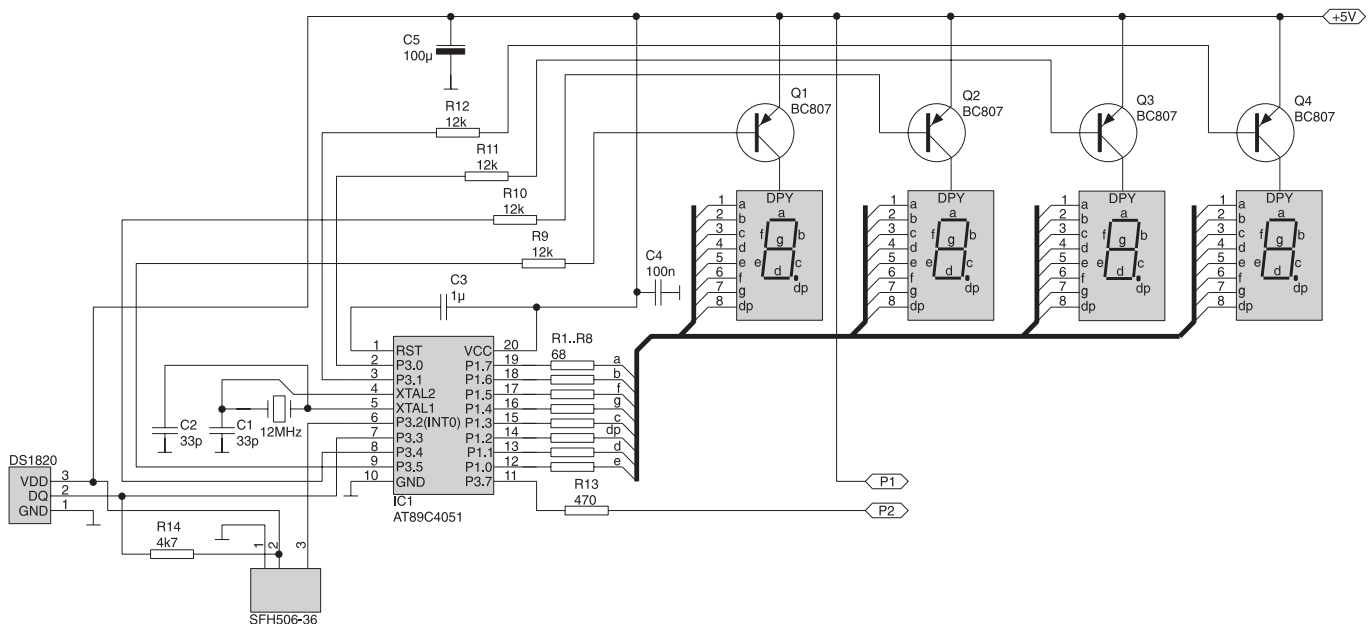
i były czynnikiem determinującym autora do poszukiwania rozwiązań bardziej „współczesnych”.

W początkowym zamyśle urządzenia miało jedynie włączać i wyłączać ogrzewacz za pomocą pilota, podobnie jak czyni się to z domowym sprzętem AV. Jednak wykorzystanie mikroprocesora umożliwiło wyposażenie sterownika w dodatkowe funkcje. Urządzenie umożliwia:

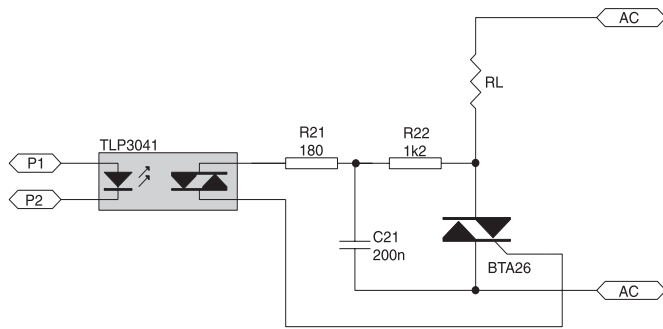
- odczyt temperatury otoczenia z rozdzielczością 0,1°C,
- odczyt temperatury minimalnej i maksymalnej w danym przedziale czasowym,
- manualne włączanie i wyłączanie, za pomocą pilota, dowolnego elektrycznego urządzenia grzewczego,

Takie sytuacje, w przypadku autora, powodowały konieczność zaprzestania wykonywanych właśnie zajęć (na ogół w najmniej stosownym momencie - np.: podczas precyzyjnego lutowania), przejścia na drugi koniec po-

koju, włączenia podgrzewacza i powrotu na miejsce. Chociaż podobna gimnastyka, z punktu widzenia higieny pracy, jest elementem bardzo pozytywnym, to jednak zbyt częste tego typu spacerów powodowały spadek efektywności pracy



Rys. 1. Schemat elektryczny regulatora temperatury



Rys. 2. Schemat elektryczny układu sterownika mocy

- automatyczna (dwupołożeniowa) regulacja temperatury pomiędzy wartością minimalną t_L i maksymalną t_H ,
- wprowadzanie za pomocą pilota wartości t_L i t_H ,
- włączanie urządzenia grzewczego „w zerze” sieci energetycznej,
- automatyczne wyłączenie urządzenia grzewczego w przypadku przekroczenia dopuszczalnego czasu pracy.

Część sprzętowa

Regulator składa się z płytki sterownika mikroprocesorowego zawierającej cztery siedmiosegmentowe wyświetlacze LED, zamontowanej bezpośrednio na obudowie typowego zasilacza wtyczkowego, oraz układu bezstykowego sterownika mocy zamkniętego w oddzielnej obudowie.

Schemat elektryczny sterownika mikroprocesorowego przedstawiono na rys. 1. Zastosowano tu popularny mikrokontroler z rodziny 51 typu AT89C4051 firmy Atmel z 4 kB pamięcią programu typu Flash. Mikrokontroler jest taktowany sygnałem zegarowym o częstotliwości 12 MHz. Jako czujnik temperatury wykorzystano cyfrowy termometr DS1820 firmy Maxim/Dallas. Zintegrowany odbiornik podczerwieni dołączony jest do wejścia przerwania zewnętrznego INT0 mikrokontrolera. Port P1 mikrokontrolera steruje bezpośrednio katodami wyświetlaczy siedmiosegmentowych. Anody wyświetlaczy sterowane są poprzez klucze tranzystorowe z odpowiednich linii portu P3 mikrokontrolera. Na płycie drukowanej regulatora zastosowano montaż przewlekany (mikrokontroler, kwarc, wyświetlacze, kondensator elektrolityczny) oraz SMT (rezystory, kondensatory, tranzystory).

Czujnik temperatury jest podłączony do płytki regulatora za pomocą kilkunastocentymetrowego przewodu. Oddalenie czujnika pomiarowego od płytki regulatora i obudowy zasilacza jest konieczne ze względu na wydzielanie ciepła przez transformator sieciowy, co w przypadku zamontowania czujnika bezpośrednio na płytce, powodowałoby błędne (zawyżone) odczyty temperatury.

Zamiast elektromagnetycznego przekaźnika wyłączającego urządzenie wykonawcze zastosowano moduł złożony z optotriaka i triaka o dużej mocy (rys. 2). Wykorzystywany optotriak posiada dodatkowo układ włączania „w zerze” sieci energetycznej.

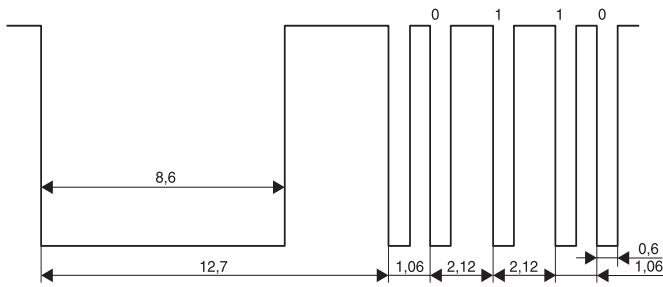
Dodatkowe elementy bierne (R21, R22, C21) łączące optotriak z triakiem są zgodne z typową aplikacją podawaną przez producenta i umożliwiają poprawne sterowanie urządzeń również o charakterze indukcyjnym. Jako element wykonawczy zastosowano triak BTA26 o typowym prądzie przewodzenia 26 A, co jest wartością aż nadto wystarczającą do sterowania urządzeń grzewczych o mocach znacznie ponad 2 kW. Triak zamocowano na radiatorze o dość kompromisowych wymiarach, dostosowanych do objętości obudowy (w przypadku sterowania urządzeń grzewczych o większych mocach konieczne jest zapewnienie odpowiednich warunków chłodzenia triaka poprzez m.in. odpowiedni dobór powierzchni radiatora).

Układ sterownika mocy zamknięto w oddzielnej obudowie z tworzywa sztucznego na której zamocowano typowe gniazdko elektryczne służące do podłączenia grzejnika.

Obsługa i działanie

Po włączeniu zasilania (włożeniu zasilacza wtyczkowego z płytką mikrokontrolera do gniazdko elektrycznego) regulator bezpośrednio wyświetla bieżącą temperaturę otoczenia. Wciśnięcie klawisza „1” w pilocie zdalnego sterowania spowoduje uruchomienie urządzenia grzewczego. Na lewym skrajnym wyświetlaczu pojawia się wówczas „wędrujący wokół” świecący segment sygnalizujący pracę urządzenia wykonawczego. Wyłączenie urządzenia wymaga wciśnięcia klawisza „2” w pilocie, lub następuje automatycznie po około 20 minutach od momentu ostatniego wciśnięcia klawisza „1”. Wciśnięcie klawisza „3” spowoduje wyświetlenie minimalnej zarejestrowanej temperatury. Sygnalizowane jest to migającym dolnym segmentem (c) lewego skrajnego wyświetlacza. Wciśnięcie w tym momencie klawisza „POWER” powoduje przepisanie bieżącej temperatury jako temperatury minimalnej (kasowanie pamięci temperatury minimalnej) - na wyświetlaczu pojawia się na parę sekund symbol „SEt”. Jeżeli klawisz „POWER” nie został wciśnięty, powrót do stanu wyświetlania temperatury bieżącej następuje automatycznie po kilkunastu sekundach lub po wciśnięciu przycisku „PHONO” (klawisz tuż obok „POWER” pełniący funkcję klawisza ESC) lub klawisza „2”. Analogicznie naciśnięcie klawisza „4” podczas wyświetlania temperatury bieżącej powoduje wyświetlenie temperatury maksymalnej, co sygnalizowane jest migającym górnym segmentem (a) pierwszego wyświetlacza. Funkcje rozkazów „PHONO”, „POWER” i „2” są analogiczne jak dla funkcji wyświetlania temperatury minimalnej. Wciśnięcie klawisza „AM” w czasie wyświetlania temperatury bieżącej spowoduje przełączenie trybu pracy regulatora z manualnego na automatyczny. Automatyczny tryb pracy sygnalizowany jest wyświetlaniem co parę sekund symbolu „Auto” na wyświetlaczu. W tym trybie bieżąca wartość temperatury porównywana jest z dwiema wartościami progowymi t_L i t_H . Jeżeli bieżąca wartość temperatury jest

niższa niż wartość progowa t_L wówczas następuje załączenie urządzenia wykonawczego. Wyłączenie urządzenia wykonawczego następuje kiedy temperatura otoczenia osiągnie wartość większą niż wartość progowa t_H lub po przekroczeniu dopuszczalnego czasu pracy. W trybie automatycznym aktywne są również klawisze „1” i „2” mające identyczne znaczenie jak w trybie manualnym. Można więc w każdej chwili klawiszem „1” załączyć urządzenie grzewcze, a klawiszem „2” wyłączyć - przełączając jednocześnie regulator w stan pracy manualnej. Przejście do trybu manualnego można również osiągnąć wciskając klawisz „FM” w pilocie zdalnego sterowania. Aby sprawdzić bieżące nastawy wartości temperatury progowych t_L i t_H należy odpowiednio nacisnąć klawisze „VOL-” lub „VOL+”. Na lewym skrajnym wyświetlaczu widnieje wtedy symbol wyświetlanej wartości: zapalone segmenty a, b, f, g dla wartości t_H oraz segmenty c, d, e, g dla wartości t_L . Powrót do wyświetlania temperatury bieżącej następuje po naciśnięciu klawisza „PHONO” lub jeżeli nie wciśnięto żadnego klawisza automatycznie po kilkunastu sekundach ale tylko wówczas gdy załączone jest urządzenie grzewcze. Jeżeli w czasie wyświetlania temperatury progowych jeszcze raz zostanie naciśnięty klawisz „VOL-” lub „VOL+” wówczas istnieje możliwość wprowadzenia nowych wartości progowych (t_L lub t_H zależnie od tego która wartość jest właśnie wyświetlana). Wygaszone zostają wówczas pozostałe wyświetlacze z wyjątkiem pierwszego pokazującego symbol wprowadzanej wartości. Naciskanie teraz klawiszy numerycznych zostanie zinterpretowane jako określenie kolejno dziesiątek, jednostek i dziesiątych części danej nastawy t_L lub t_H . W każdej chwili tryb wprowadzania danych można opuścić naciskając klawisz „PHONO”. W przypadku kiedy uruchomiony jest ogrzewacz wyjście z tego trybu może nastąpić również automatycznie po kilkunastu sekundach od chwili ostatniego wciśnięcia dowolnego klawisza w pilocie zdalnego sterowania. Zatwierdzenie wprowadzonej wartości odby-



Rys. 3. Fragment przebiegu na wyjściu odbiornika podczerwieni (wartości w ms)

wa się poprzez naciśnięcie klawisza „POWER“. Dodatkowo sprawdzana jest spójność wprowadzanych danych. Jeżeli $t_H - t_L < 0,4^{\circ}\text{C}$ wówczas sygnalizowany jest błąd (symbol „Err“ na wyświetlaczu) i wprowadzane wartości nie są zapamiętywane. Jeżeli wprowadzone dane spełniają powyższe równanie i dodatkowo jeżeli wartości t_L i t_H mieszczą się w przedziale $0...25,5^{\circ}\text{C}$ (dane można zapisać za pomocą 8 bitów), wówczas przepisywane są do nieulotnej pamięci E²RAM znajdującej się w układzie DS1820 (wykorzystywane są dwa bajty użytkownika). Jeżeli któraś z wartości t_L lub t_H przekracza $25,5^{\circ}\text{C}$ wówczas nie jest zapisywana w pamięci nieulotnej lecz jedynie w pamięci RAM mikrokontrolera. Dane z nieulotnej pamięci układu DS1820 przepisywane są do pamięci RAM mikrokontrolera inicjując wartości t_L i t_H podczas sekwencji startowej po włączeniu zasilania. Jeśli wystąpi błąd odczytu z pamięci nieulotnej sygnalizowane jest to komunikatem „Err“ na wyświetlaczu tuż po włączeniu zasilania i obie wartości t_L i t_H domyślnie inicjowane są zerami (wartości te można zmienić wciskając dwukrotnie klawisze „VOL-“ lub „VOL+“).

Oprogramowanie

Oprogramowanie sterujące regulatorem wykonano w języku C wykorzystując kompilator Keil w bardzo starej, ale w pełni funkcjonalnej, wersji 3.20 dla DOS-a. Jedynie procedury obsługi magistrali jednoprzewodowej dla układu czujnika temperatury zaimplementowano w assemblerze. Całość kodu wynikowego (po linkowaniu) zajmuje prawie 4 kB pamięci programu mikrokontrolera. W dalszej części artykułu zostanie skrótowo

opisane dekodowanie sygnałów zdalnego sterowania oraz procedury obsługi magistrali jednoprzewodowej i ich zastosowanie do komunikacji z układem DS1820.

Dekodowanie sygnałów zdalnego sterowania

Jako pilot zdalnego sterowania wykorzystano model RC-200S firmy TEAC. Analiza oscyloskopem cyfrowym generowanych przez pilota sygnałów wykazała, iż pracuje on w standardzie NEC80 (w tym samym standardzie pracuje też np. pilot od karty tunera TV PixelView, lecz pilot ten inaczej koduje poszczególne klawisze). Oznacza to, że czas trwania impulsu (poza nagłówkiem) jest w tym standardzie stały, zaś kodowanie „0“ i „1“ realizowane jest poprzez różny czas przerwy między impulsami. W standardzie NEC80 przesyłane są 32 bity danych.

Szczegółowe informacje na temat kodowania i sposób dekodowania sygnałów zdalnego sterowania zamieszczono w EP12/2002. Sposób dekodowania sygnałów zdalnego sterowania zaproponowany w tym artykule jest nieco inny (powstał zanim ukazał się wspomniany numer EP).

Na rys. 3 przedstawiono fragment przebiegu otrzymanego na wyjściu odbiornika podczerwieni podczas nadawania wybranego rozkazu. Dekodowanie poszczególnych rozkazów będzie więc polegało na pomiarze czasu pomiędzy kolejnymi opadającymi zboczami sygnału z odbiornika podczerwieni. Opadające zbocze sygnału z odbiornika podczerwieni, podłączonego do wejścia przerwania zewnętrznego mikrokontrolera, spowoduje żądanie przyjęcia przerwania. Aby przyjęcie przerwania było możliwe konieczne jest wcześniejsze

ustawienie bitu IT0 w słowie sterującym TCON - oznaczające sposób przyjęcia przerwania opadającym zboczem na wejściu INTO - oraz ustawienie bitów EX0 i EA w masce przerwania IE. Pomiar czasu pomiędzy kolejnymi opadającymi zboczami na wejściu INTO będzie się odbywał za pomocą sprzętowego licznika T0 pracującego w trybie 1. Na list. 1 przedstawiono kod źródłowy funkcji w języku C obsługującej przerwanie z wejścia zewnętrznego i realizującej dekodowanie poszczególnych rozkazów oraz funkcji obsługi przerwania z licznika T0 (przerwanie zgłaszane w momencie przepelnienia licznika).

Bity *start_seq* i *start_bit* wykorzystywane są do sygnalizacji kolejno rozpoczęcia nadawania rozkazu (nagłówka) i wystąpienia bitu startu. Ustawienie bitu *rdy* oznacza, że odebrano pomyślnie rozkaz, którego kod znajduje się w 32-bitowej zmiennej typu *long* o nazwie *byte*. Typ danych bit nie jest typem standardowym charakterystycznym dla języka C. W kompilatorze Keil jest to bit oznaczający 8-bitowy adres bitu w pamięci RAM mikrokontrolera. Zmienna *mh* służy do tymczasowego przechowywania starszego bajtu licznika T0. Od tej też operacji - zapamiętania stanu bajtu TH0 licznika T0 - rozpoczyna działanie funkcja obsługi przerwania zewnętrznego. Następnie zerowany jest licznik T0. Jeżeli przyjęte zostało przerwanie, a bit *start_seq* nie jest ustawiony, oznacza to, że rozpoczyna się nadawanie rozkazu. Ustawiany jest wówczas bit *start_seq* i funkcja obsługi przerwania kończy działanie. Przy kolejnym wywołaniu funkcji (kolejnym przerwaniu) sprawdzany jest stan bitu *start_bit* (*start_seq* jest już ustawiony). Jeżeli nie był on zapalony oznacza to prawdopodobieństwo wystąpienia bitu startu. Sprawdzany jest teraz czas, który upłynął od ostatniego przerwania. Przy czym każdy pomiar czasu odbywa się z dokładnością do 256µs - brany jest pod uwagę jedynie starszy bajt licznika T0. Jeżeli czas ten zawiera się w przedziale od 11,5 ms do

List. 1. Kod źródłowy funkcji obsługi przerwania i dekodowania rozkazów

```
void int_ext0(void) interrupt 0 using 1
{
    mh=TH0;    TL0=0; TH0=0;

    if (!start_seq)
        start_seq=1;
    else
        if (!start_bit)
        {
            if (mh>45&&mh<54)
                { start_bit=1; bitcnt=0; byte=0;
                  TR1=0; ET1=0; P1=255;
                }
            else
                start_seq=0;
        }
    else
        if (start_bit)
        {
            if (mh>12) { start_seq=start_bit=0; ET1=1; return; }
            if (bitcnt<32)
                {
                    byte<<=1;
                    if(mh>6) byte|=1;
                }
            bitcnt++;
            if (bitcnt>32)
                {
                    rdy=1;
                    start_bit=start_seq=0;
                    TR1=1; ET1=1;
                }
        }
}

void int_t0(void) interrupt 1 using 1
{
    ET1=1; TR1=1;
    start_bit=start_seq=0;
}

Funkcje wykorzystują następujące zmienne zewnętrzne (globalne):

bit start_bit,start_seq,rdy;
unsigned char mh,bitcnt;
unsigned long byte;
```

14 ms oznacza to, że odebrany został nagłówek i bit startu. Stąd ustawiany jest teraz bit *start_bit*, zerowany licznik bitów *bitcnt* i zerowana zmienna *byte*. Dodatkowo blokowany jest licznik T1 sterujący wyświetlaczem i wszystkie bity portu P1 ustawiane są w stan wysoki, powodując wygaszenie wyświetlacza. Przy kolejnym zgłoszeniu przerwania również sprawdzany jest czas, który upłynął od ostatniego przyjęcia przerwania. Jeżeli jest on większy niż 3 ms oznacza to błąd i procedura dekodowania rozkazów rozpoczyna się od początku. W przeciwnym przypadku, o ile nie odebrano jeszcze 32 bitów, wszystkie bity zmiennej *byte* przesuwane są o jedną pozycję w lewo i jeżeli wyżej wymieniony czas jest większy niż 1,5 ms - co oznacza że odebrany bit ma wartość 1 - ustawiany jest najmłodszy bit zmiennej *byte*. Następnie zwiększany jest

o jeden licznik bitów. Jeżeli przy kolejnej obsłudze przerwania odebrano już 32 bity (*bitcnt > 32*) oznacza to wystąpienie bitu stopu i kompletne odebranie rozkazu. Ustawiana jest wówczas globalna flaga *rdy* sygnalizująca że zmienna *byte* zawiera kod odebranego rozkazu. Jeżeli z jakiegoś powodu transmisja zostanie przerwana (zakłócona) przed odebraniem 32 bitów, wówczas najpóźniej po 65,5 ms nastąpi przepełnienie licznika T0 i wysłane zostanie żądanie przyjęcia przerwania od tego licznika. Funkcja obsługi tego przerwania (*int_t0()*) zeruje bity *start_bit* i *start_seq* powodując, że przy kolejnym przerwaniu z odbiornika podczerwieni, procedura dekodowania rozkazów rozpocznie się od początku.

Wykorzystanie tak zdefiniowanych funkcji dekodowania rozkazów z odbiornika zdalnego sterowania w programie głównym (programie użytkownika) wymaga zade-

klarowania wymienionych wcześniej zmiennych na zewnątrz wszystkich funkcji i wygląda następująco:

```
if(rdy)
// zdekodowano rozkaz
{ rdy=0;
  if(byte==0xe19e31ce)
  { // wciśnięto klawisz
    // "1" w pilocie
  } else
  if(byte==0xe19eb14e)
  { // wciśnięto klawisz
    // "2" w pilocie
  }...
}
```

Lub też używając krótszych - 16 bitowych (dwu bajtowych) stałych, z użyciem wskaźników, tak jak w programie obsługi regulatora:

```
int *ptr;
//deklaracja wskaźnika
//do zmiennych typu int
...
ptr=(int*)&byte;
//wskaźnik ptr zawiera
//adres zmiennej byte
if(rdy)
// zdekodowano rozkaz
{ rdy=0;
  if(*ptr++==0xe19e)
  //stały kod
  //dla danego pilota
  {
    if(*ptr==0x31ce)
    { // wciśnięto klawisz
      // "1" w pilocie
    } else
    if(*ptr==0xb14e)
    { // wciśnięto klawisz
      // "2" w pilocie
    } ...
  }
}
```

Ponieważ zmienna *byte* jest obiektem 32-bitowym typu *long* a wskaźnik *ptr* zgod-

WYKAZ ELEMENTÓW

Rezystory

R1...R8: 68Ω
R9...R12: 12kΩ
R13: 470Ω
R14: 4,7kΩ
R21: 180Ω
R22: 1,2kΩ

Kondensatory

C1, C2: 33pF,
C3: 1μF,
C4: 0,1μF,
C5: 100μF,
C21: 0,22μF

Półprzewodniki

Wyświetlacze 7-segmentowe LED - 4szt,
Mikroprocesor AT89C4051,
Tranzystory BC807(SMD) - 4 szt,
Termometr DS1820,
Scalony odbiornik podczerwieni,
Optotriak TLP3041,
Triak BTA26

Różne

Rezonator kwarcowy 12MHz,
Zasilacz "wtyczkowy" 5V, 200mA,
Pilot zdalnego sterowania

nie z definicją może pokazywać na obiekty 16-bitowe typu *int* podczas przypisania adresu konieczna jest konwersja typów z *long** do *int**. W przypisaniu *ptr=(int*)&byte* zastosowano jawną konwersję typów z użyciem operatora rzutowania (*int**). Dzięki takiemu zabiegowi wskaźnik *ptr* pokazuje na starszych 16 bitów zmiennej *byte* zaś *ptr+1* na młodszych 16 bitów tej zmiennej. Fakt ten wykorzystano w prezentowanym programie.

Zbigniew Hajduk