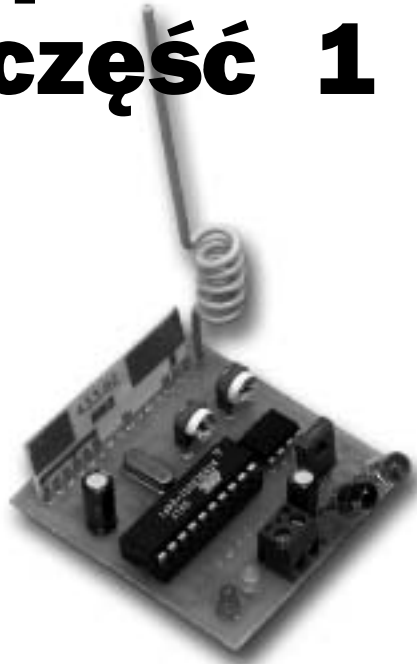


# Radiowy przedłużacz pilotów, część 1

## AVT-559

*Pomysł budowy „przedłużacza” zrodził się, gdy autor, utrudzony całodzienną pracą, musiał wstać z wygodnego fotela stojącego przed komputerem (służącym w tym przypadku jako odbiornik - monitor telewizyjny) i pójść do sąsiedniego pokoju, aby zmienić kanał w tunerze satelitarnym.*



Oczywiście, aby zmienić kanał w tunerze satelitarnym stojącym w innym pokoju, można wykorzystać jeden z „przedłużaczy”, opisanych chociażby na łamach Elektroniki Praktycznej czy Elektroniki dla Wszystkich. Jednak prezentowane w tych artykułach układy wymagały albo przeciągania przewodów z pomieszczenia do pomieszczenia, albo przynajmniej wykorzystania tego samego egzemplarza pilota zdalnego sterowania. Do sterowania tunerem z innego pomieszczenia autor postanowił wykorzystać inny egzemplarz pilota niż oryginalny (ten ostatni pozostawiając w pokoju z tunerem), gdyż jak wynika z doświadczenia, przemieszczanie pilota z miejsce na miejsce uniemożliwia jego odnalezienie w momencie, gdy jest najbardziej potrzebny.

Zaprojektowany układ ma następujące cechy:

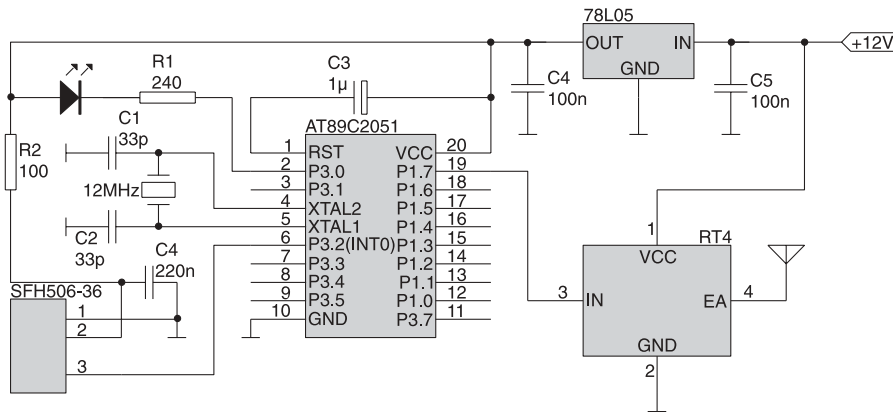
- możliwość wykorzystania dowolnego pilota pracującego w standardzie RC5,
- radiowa transmisja danych pomiędzy odbiornikiem i nadajnikiem podczerwieni,
- programowa regeneracja impulsów po stronie odbiornika radiowego (nadajnika podczerwieni), poprawiająca niezawodność

- radiowej transmisji danych,
- kontrola poprawności danych (CRC) przesyłanych kanałem radiowym,
- prosta wizualizacja pracy układów (sygnalizacja odebrania polecenia z pilota i wysłania pakietu danych drogą radiową, sygnalizacja odebrania danych z kanału radiowego, itp.).

### Opis układów nadawczo-odbiorczych - część sprzętowa

Cały zestaw składa się z dwóch płytek, zawierających repeatery sygnałów zdalnego sterowania, wykorzystujące różne media transmisyjne: podczerwień i fale radiowe.

Pierwszy przemiennik IR-RF (podczerwień - fale radiowe) dekoduje sygnały z pilota zdalnego sterowania pracującego w standardzie RC5 i uzyskaną w ten sposób informację retransmituje drogą radiową do drugiego przemiennika RF-IR (fale radiowe - podczerwień). Przemiennik ten odbiera, dekoduje i odpowiednio przetwarza informację (m.in. zamienia



Rys. 1. Schemat przemiennika IR-RF

adres urządzenia RC5, zamienia niektóre kody rozkazów), a następnie ponownie koduje zgodnie ze standardem RC5 i przesyła do modulatora (nadajnika) podczerwieni. Dzięki zastosowaniu transmisji radiowej pomiędzy przemiennikami, jest możliwe przesyłanie sygnałów zdalnego sterowania na znacznie większe odległości niż w przypadku zastosowania podczerwieni jako medium transmisyjnego. Odległość ta zależna jest od parametrów toru radiowego oraz od warunków otoczenia.

Schemat układu przemiennika IR-RF przedstawiono na rys. 1. Jako jednostkę centralną zastosowano mikrokontroler AT89C2051. Do odbioru sygnałów z pilota zdalnego sterowania wykorzystano scalony odbiornik podczerwieni. Jako nadajnik radiowy zastosowano hybrydowy moduł RT4 firmy *Telecontrolli* pracujący ze stosunkowo niewielką mocą (max. 10 dBm) na częstotliwości 433.92 MHz. Nadajnik pracuje z modulacją ASK.

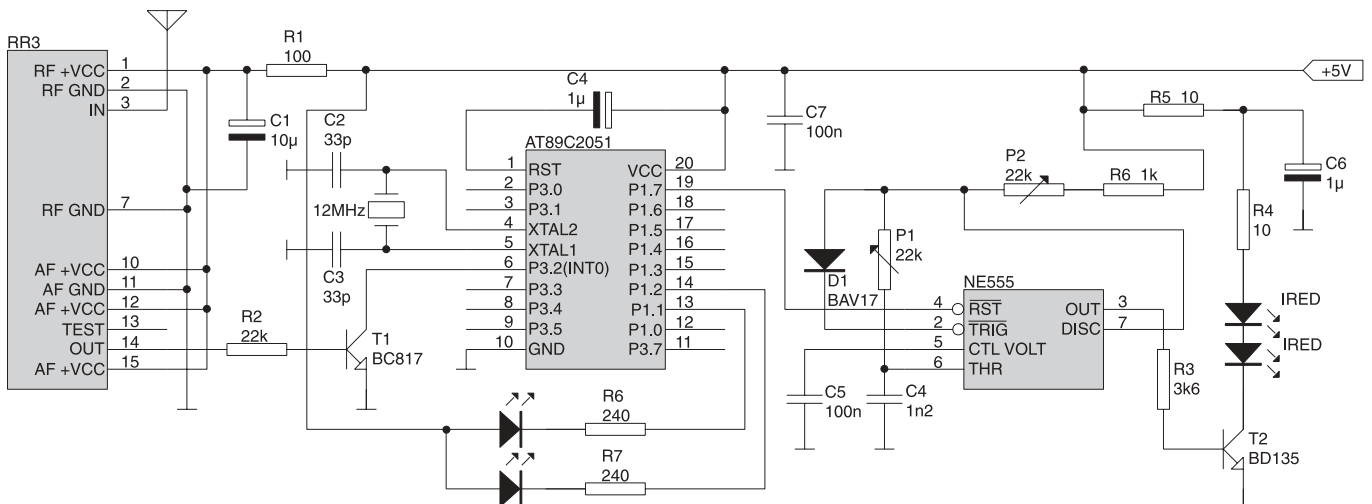
Sposób kodowania sygnałów dla nadajnika zostanie opisany dalej. Komentarza wymaga sposób podłączenia wejścia danych nadajnika, które jest sterowane bezpośrednio z portu P1.7 mikrokontrolera. Sam nadajnik zasilany jest napięciem 12 V (w celu uzyskania większej mocy wyjściowej), jednak zgodnie z dokumentacją do wysterowania jego wejścia danych wystarcza sygnał o amplitudzie 5 V. Napięcie o takiej (mniej więcej) amplitudzie dostępne jest bezpośrednio na linii portu mikrokontrolera, dlatego nie ma potrzeby stosowania dodatkowych układów przesuwania poziomu. Przemiennik IR-RF zasilany jest napięciem 12 V z niewielkiego zasilacza wtyczkowego.

Układ drugiego przemiennika RF-IR - jest nieco bardziej skomplikowany. Jego schemat zamieszczono na rys. 2. Jako odbiornik radiowy zastosowano superreakcyjny, hybrydowy moduł RR3 firmy *Telecontrolli*. Dane z modułu odbiornika radiowego są negocwa-

ne przez tranzystor T1 i podawane na wejścia przerwania mikrokontrolera AT89C2051. Część układu złożona z multiwibratora astabilnego NE555, tranzystora T2 oraz dwóch diod IRED jest modulatorem podczerwieni wytwarzającym falę nośną o częstotliwości 36 kHz. Modulator sterowany jest sygnałem z portu P1.7 mikrokontrolera. Poziom niski na tym wyjściu mikrokontrolera wprowadza wejście zerowania timera 555 w stan aktywny, powodując wymuszenie poziomu niskiego na wyjściu i brak wysterowania tranzystora T2 i diod IRED. Jeżeli na tymże wejściu timera pojawi się poziom wysoki, wówczas układ generuje falę prostokątną o częstotliwości i współczynniku wypełnienia regulowanym za pomocą potencjometrów P1 i P2. Diodę D1 dodano w celu uzyskania możliwości zmiany współczynnika wypełnienia w pełnym zakresie (bez diody D1 można jedynie uzyskać wypełnienie >50%). Tranzystor T2 pracuje jako klucz, bezpośrednio sterujący diodami IRED. Z portów P1.1 i P1.2 sterowane są diody LED sygnalizujące stan pracy przemiennika. Cały układ zasilany jest napięciem 5 V z zasilacza wtyczkowego.

### Działanie przedłużacza radiowego

W wersji pierwotnej radiowy przedłużacz pilotów akceptuje sygnały pilotów, których adres urządzenia RC5 jest równy 00h (pilot od OTV Elemis), zamieniając go po stronie przemiennika



Rys. 2. Schemat przemiennika RF-IR

RF-IR na adres 16h (pilot od tunera satelitarnego). Dodatkowo radiowy przedłużacz pilotów dokonuje zamiany niektórych kodów rozkazów, tak aby układy klawiatury obu pilotów były zgodne. W przypadku poprawnego odebrania polecenia RC5 w przemienniku IR-RF, na krótko zapala się dioda LED sygnalizująca ten fakt. Jeżeli adres urządzenia RC5 jest różny od 00h, wówczas dioda świeci się znacznie dłużej (kod polecenia nie jest transmitowany do drugiego przemiennika RF-IR). Pomyślne odebranie danych z toru radiowego sygnalizowane jest krótkim mignięciem zielonej diody LED w przemienniku RF-IR. W przypadku błędu w transmisji (nieprawidłowe CRC) zapalana jest dioda czerwona.

### Oprogramowanie (wybrane zagadnienia)

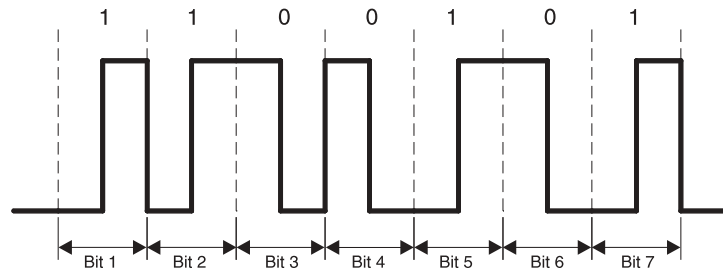
Część sprzętowa radiowego przedłużacza pilotów, zawierająca typowe komponenty, jest jak widać bardzo prosta. Właściwe cechy funkcjonalne nadaje przedłużaczowi oprogramowanie. Przygotowano je w języku C i skompilowano kompilatorem firmy KEIL.

### Dekodowanie sygnałów zdalnego sterowania w standardzie RC5

Ponieważ zarówno pilot od tunera satelitarnego, jak i drugi pilot przeznaczony do sterowania tunerem z innego pomieszczenia (wykorzystano pilot od telewizorów ELEMIS) pracowały w standardzie RC5, stąd zaistniała konieczność dekodowania sygnałów zdalnego sterowania opartych na tym standardzie. W artykule [2] opisano zbiór gotowych funkcji w języku C służących do dekodowania sygnałów zdalnego sterowania. Jednak wykorzystałem własne procedury oparte na bardziej efektywnym algorytmie bazującym na automacie sekwencyjnym [3].

**Tab. 1. Czasy występowania zdarzeń podczas transmisji RC5**

Zdarzenie	Ideal	Min.	Max.
ki	889	444	1333
kp	889	444	1333
di	1778	1334	2222
dp	1778	1334	2222



Rys. 3. Kodowanie RC5

Kod RC5 charakteryzuje się jednakowym czasem trwania wszystkich bitów. Zmiana poziomu sygnału (zbocze narastające lub opadające) w połowie przedziału czasowego przypisanego każdemu bitowi określa logiczną wartość. I tak „0” jest kodowane jako opadające zbocze sygnału, a „1” jako zbocze narastające (więcej na temat kodu RC5 można znaleźć w [1,2,3,4]).

Na rys. 3 przedstawiono ilustrację sposobu kodowania bitów w standardzie RC5. Przesyłanych jest 14 bitów, z których dwa pierwsze są bitami startowymi (dwie jedynki), następnie przesyłany jest bit informujący o przytrzymaniu klawisza w pilocie, następujących 5 bitów jest adresem urządzenia RC5, a ostatnie 6 bitów stanowi kod transmitowanego polecenia.

Algorytm dekodowania sygnałów w standardzie RC5 można zapisać za pomocą grafu automatu sekwencyjnego [3], czyli inaczej grafu przejść i wyjść według modelu Mealy'ego, znanego z teorii układów cyfrowych. Graf taki przedstawiono na rys. 4. Etykiety opisujące łuki oznaczają odpowiednio rodzaj zdarzenia (impuls - przerwa) oraz po przecinku wartość logiczną zdekodowanego bitu. Zdarzenia opisane są następującymi symbolami:

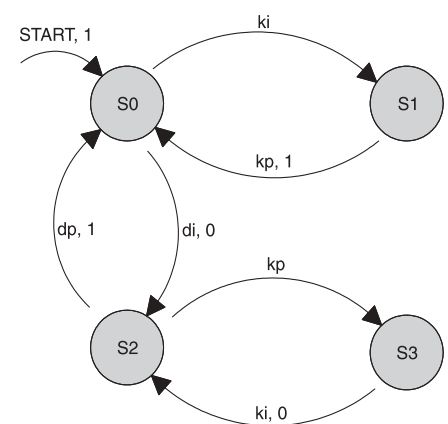
- ki - krótki impuls,
- kp - krótka przerwa,
- di - długi impuls,
- dp - długa przerwa,

przy czym co do wartości bezwzględnej zdarzenia te - zgodnie ze standardem - sklasyfikowane są w sposób pokazany w tab. 1 (czas podany w mikrosekundach).

Algorytm opisany grafem z rys. 4 rozpoczyna swoje działanie od stanu S0, przy czym wejście do tego stanu spowodowane jest pojawieniem się zbocza narastające-

go dekodowanego sygnału i jednocześnie oznacza, że odebrany został pierwszy bit o wartości logicznej 1. Następnie dekodowanych jest pozostałych 13 bitów (odebranie kolejno 14 bitów stanowi warunek zatrzymania pracy automatu). Każde występujące zdarzenie nieopisane grafem z rys. 4 traktowane jest jako błąd, powodujący przerwanie dekodowania i ponowne jego rozpoczęcie od punktu startowego.

Na list. 1 przedstawiono funkcję obsługi przerwania zewnętrznego, dokonującą dekodowania sygnałów zdalnego sterowania w standardzie RC5 zgodnie z algorytmem opisanym diagramem stanów z rys. 4. Funkcja wykorzystuje sprzętowy licznik T0 mikrokontrolera pracujący w trybie 1, taktowany z częstotliwością 1 MHz. Do klasyfikacji zdarzeń z diagramu wykorzystano tylko bardziej znaczący bajt licznika T0. Zastosowano też nieco inne (bardziej rygorystyczne) niż podane w tabeli wartości progowe definiujące impuls (przerwę) długi i krótki (impuls krótki zawiera się w przedziale 768μs...1279μs, impuls długi w 1280μs...2047μs).



Rys. 4. Diagram stanów opisujący algorytm dekodowania sygnałów RC5

List. 1. Funkcja obsługi przerwania zewnętrznego służąca do dekodowania sygnałów zdalnego sterowania

```

void int_ext0(void) interrupt 0 using 1
{
    unsigned int tmp;
    char p_th,state,cntbit;
    bit p_in,c_in;

    TL0=0; TH0=0; tmp=0; cntbit=1; state=0;

    while(1)
    {
        p_in=in; //zapamiętanie wejścia
        while(p_in==in) if (TH0>8) return;
        //oczekiwania na zbocze, jeżeli zbyt długo powrót (błąd)

        p_th=TH0;
        // zapamiętanie stanu licznika T0 (tylko starszy bajt)

        TL0=0; TH0=0; c_in=in;
        // zerowanie licznika T0 i buforowanie wejścia

        switch(state)
        {
            case 0: // stan S0
                if(!p_in&&c_in&&p_th>2&&p_th<5) state=1;
                // jeżeli krótki impuls przejdź do stanu S1
                else
                if(!p_in&&c_in&&p_th>4&&p_th<8)
                { state=2; tmp<<=1; cntbit++; }
                // jeżeli długi impuls to następny stan - S2,
                // odebrany bit ma wartość 0

                else return;
                // w przeciwnym przypadku błąd

                break;

            case 1: // Stan S1
                if(p_in&&c_in&&p_th>2&&p_th<5)
                { state=0; tmp<<=1; tmp|=1; cntbit++; }
                // jeżeli krótka przerwa następny stan - S0,
                // odebrany bit ma wartość 1

                else return;
                break;

            case 2: // Stan S2
                if(p_in&&c_in&&p_th>4&&p_th<8)
                { state=0; tmp<<=1; tmp|=1; cntbit++; }
                // Jeżeli długa przerwa powrót do stanu S0,
                // odebrany bit ma wartość 1

                else
                if(p_in&&c_in&&p_th>2&&p_th<5) state=3;
                // Jeżeli krótka przerwa następny stan - S3

                else return;
                break;

            case 3: // Stan S3
                if(!p_in&&c_in&&p_th>2&&p_th<5)
                { state=2; tmp<<=1; cntbit++; }
                // Jeżeli krótki impuls następny stan - S2,
                // odebrany bit ma wartość 0

                else return;
                break;
        }
        if(cntbit>=14) { rdy=1; addr=(tmp>>6)&0x1f; cmd=tmp&0x003f;
                    T=(tmp&0x0800)?1:0; return; }
        // Jeżeli odebrano 14 bitów - koniec dekodowania
    }
}

```

Funkcja komunikuje się z programem głównym poprzez następujące zmienne:

```

bit rdy,T;
char addr,cmd;

```

Poprawne odebranie transmisji z pilota zdalnego sterowania sygnalizowane jest ustawieniem globalnego bitu *rdy*. Wówczas zmienne *addr* oraz *cmd* zawierają odpowiednio adres urządzenia RC5 oraz kod polecenia. Zmienna bitowa *T* przechowuje wartość bitu świadczącego o przytrzymaniu klawisza. Wykorzystanie tej funkcji we własnym programie może wyglądać jak pokazano na list. 2 (przy czym należy pamiętać

o ustawieniu trybu 1 pracy licznika T0 oraz o zezwoleniu przyjmowania przerwania z zewnętrznego wejścia INT0 opadającym zboczem sygnału - wyjście ze scalonego odbiornika podczerwieni jest negowane, aktywny poziom niski).

Opisana funkcja dekodowania sygnałów w standardzie RC5 może być również wykorzystana dla innego typu mikrokontrolera, choć może się to wiązać np. ze zmianą odpowiednich stałych definiujących określone zdarzenie (inna częstotliwość pracy licznika), zmianą nazw zmiennych odnoszących się do licznika sprzętowego

List. 2. Przykład wykorzystania funkcji dekodowania kodu RC5 w programie

```

if(rdy)
{ // odebrano polecenie
  rdy=0;
  if(addr==0x16)
  { // jeżeli zgadza się adres RC5
    if(cmd==0x10)
    { // przykładowe polecenie
      ...
    }
    else
    if(cmd==0x11)
    { // kolejne polecenie
      ...
    }
    // itd.
  }
}
}

```

itp. Funkcja nie musi być też funkcją obsługi przerwania zewnętrznego, lecz wówczas należy w programie głównym umieścić fragment kodu sprawdzający cyklicznie stan wejścia sygnału z odbiornika RC5 i w przypadku kiedy wykryte zostanie opadające zbocze, należy wywołać opisaną wyżej funkcję.

**Zbigniew Hajduk**

*Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: [pcb.ep.com.pl](http://pcb.ep.com.pl) oraz na płycie CD-EP2/2004B w katalogu PCB.*

## WYKAZ ELEMENTÓW

### Rezystory

R1, R27, R28: 240Ω  
 R2, R21: 100Ω  
 R22: 22kΩ  
 R23: 3,6kΩ  
 R24, R25: 10Ω  
 R26: 1kΩ  
 P1, P2: potencjometry montażowe 22kΩ

### Kondensatory

C1, C2, C22, C23: 33pF  
 C3, C24, C28: 1μF  
 C4, C5, C25, C27: 100nF  
 C6: 220nF  
 C21: 10μF  
 C26 1,2nF

### Półprzewodniki

mikroprocesor AT89C2051 - 2szt.  
 T1: tranzystor BC817 (SMD)  
 T2: tranzystor BD135  
 scalony odbiornik podczerwieni  
 D1: dioda BAV17,  
 diody IRED - 2szt.  
 diody LED - 3szt.

### Różne

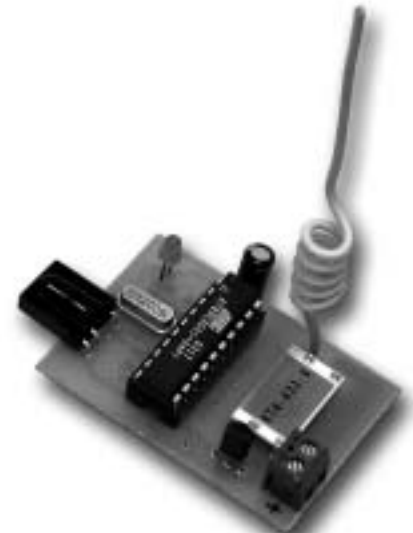
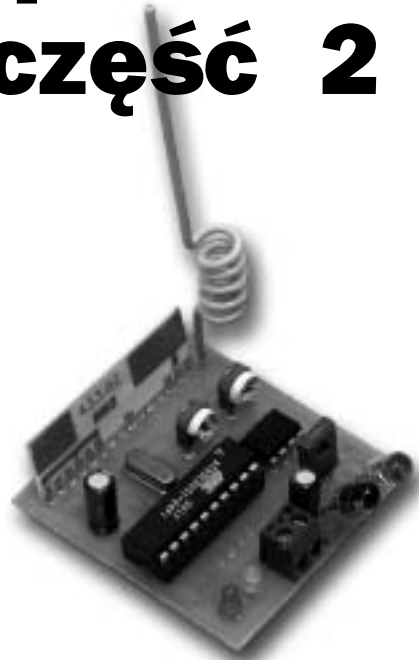
RT4: hybrydowy nadajnik 433,92MHz  
 RR3: hybrydowy odbiornik

# Radiowy przedłużacz pilotów, część 2

## AVT-559

Pomysł budowy „przedłużacza” zrodził się, gdy autor, utrudzony całodzienną pracą, musiał wstać z wygodnego fotela stojącego przed komputerem (służącym w tym przypadku jako odbiornik - monitor telewizyjny) i pójść do sąsiedniego pokoju, aby zmienić kanał w tunerze satelitarnym.

**Rekomendacje:** przyda się na pewno, jeśli zachodzi potrzeba zdalnego przełączania urządzeń sterowanych pilotem.



### Kodowanie sygnałów w standardzie RC5

Głównym zadaniem przemienika RF-IR jest wygenerowanie odpowiedniego ciągu impulsów skorelowanych czasowo, zawierających pełną informację w standardzie RC5 (nadanie rozkazu RC5). Zadanie to realizuje funkcja pokazana na **list. 3**.

Jest to funkcja obsługi przerwania z licznika T1. Licznik pracuje również w trybie 1. Przerwanie od tego licznika generowane jest co ok. 890µs (nadanie odpowiednich wartości zmiennym TL1 i TH1), czyli tyle ile wynosi połowa czasu trwania bitu w standardzie RC5. Wykorzystano tu klasyczny algorytm działający zgodnie z definicją, czyli generowanie odpowiedniego zbocza (narastającego lub opadającego) w połowie czasu trwania bitu - w zależności od wartości logicznej bitu przeznaczanego do zakodowania.

Funkcja komunikuje się z programem głównym (innymi funkcjami) poprzez następujące zmienne zewnętrzne:

```
bit p_out, hlf, val;
char bitcnt;
```

```
unsigned int rc5code;
```

Inicjowanie transmisji kodu RC5 z własnego programu polega na nadaniu wartości początko-

wych odpowiednim zmiennym i uruchomieniu licznika T1 (bit TR1), czyli należy wykonać następującą sekwencję poleceń:

```
bitcnt=0; p_out=0; hlf=1;
```

```
TR1=1;
```

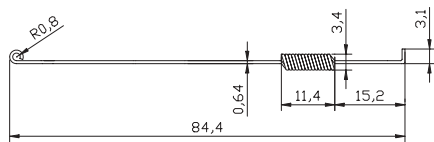
```
// while (TR1);
```

przy czym zmienna *rc5code* powinna zawierać kompletny kod rozkazu RC5 (wszystkie 14 bitów). Przed uruchomieniem licznika T1 może być też wskazane zabronienie przyjmowania innych przerw (np. przerwania zewnętrzne). Po zakończeniu wysyłania rozkazu RC5 funkcja automatycznie zatrzymuje licznik T1 i ten fakt można wykorzystać (stan bitu TR1) w celu określenia w programie zakończenia nadawania roz-

List. 3. Funkcja generowania ciągu impulsów w standardzie RC5

```
void int_t1(void) interrupt 3 using 1
{
    TL1=132; TH1=252;
    out=p_out;
    if(hlf)
    {
        val=(rc5code&0x2000)?1:0;
        if(val) p_out=0; else p_out=1;
        hlf=0; rc5code<<=1; bitcnt++;
        if(bitcnt>14) p_out=0;
    }
    else
    {
        if(val) p_out=1; else p_out=0;
        hlf=1;
        if(bitcnt>14) { p_out=0; TR1=0; }
    }
}
```

## Radiowy przedłużacz pilotów



Rys. 5. Budowa anten zastosowanych w przedłużaczu

kazu - np. w sposób pokazany na list. 3 (instrukcja *while* w linii stanowiącej komentarz).

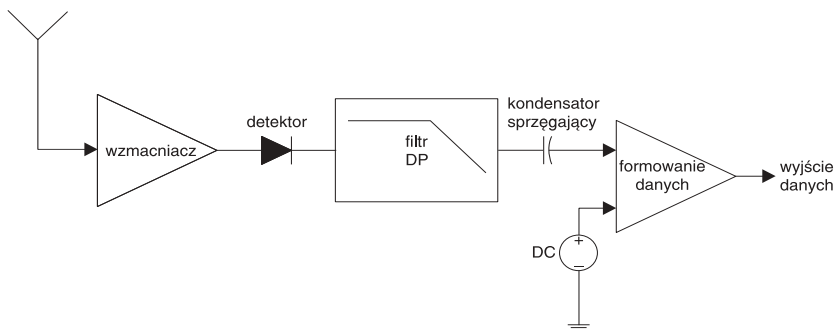
Dla obu pokazanych wyżej listingów zmienne bitowe *in* oraz *out* odnoszą się bezpośrednio do odpowiednich portów mikrokontrolera i ich definicja dla schematów z rysunków 1 i 2 powinna być następująca:

```
sbit in=P3^2;
sbit out=P1^7;
```

## Radiowa transmisja danych

Jako elementy toru radiowego (nadajnik, odbiornik) wykorzystano gotowe, stosunkowo tanie i łatwo dostępne, hybrydowe moduły firmy *Telecontrolli*. Niestety na swoich stronach internetowych firma zamieszcza bardzo skromną informację na temat produkowanych przez siebie modułów, stąd większość informacji zaczerpnięto z not aplikacyjnych firmy RFM [9,10].

Jako anteny dla przemienników zastosowano [7,8] w „skróconej” wersji anteny ćwierćfalowe. Anteny takie buduje się, dodając indukcyjność w pobliżu podstawy (kompensacja reaktancji pojemnościowej) anteny. Praktycznie polega to na uformowaniu w postaci cewki części odcinka przewodu tworzącego antenę (rys. 5). Długość odcinka od podstawy anteny do początku uformowanych zwojów cewki powinna wynosić 15,2mm. Długość samej cewki, liczącej 15 zwojów nawiniętych na średnicę 2,6mm, powinna wynosić 11,4mm. Cała antena winna mieć długość około 84mm [11]. Jako materiał na antenę należy zastosować izolowany przewód (najlepiej srebrzanek) 22AWG (średnica 0,64mm). Tak zbudowana antena ma nieco tylko mniejszy zysk energetyczny (3..4dB) w porównaniu ze zwykłą anteną ćwierćfalową, jednak charakteryzuje się ok. połowę mniejszymi rozmiarami. Można też zastosować klasyczną antenę ćwierćfalową, zbudowaną z odcin-



Rys. 6. Przetwarzanie danych w odbiorniku

ka przewodu o długości określonej przybliżonym wzorem [7]:  $7500/f$ , gdzie  $f$  jest częstotliwością pracy anteny w MHz, a długość podana jest w centymetrach. Dla częstotliwości 433.92 długość ta powinna wynosić około 17,3mm.

Na rys. 6 przedstawiono schemat toru przetwarzania danych w odbiorniku radiowym - typowy dla większości systemów radiokomunikacyjnych [10] (w tym również dla zastosowanego w projekcie odbiornika RR3 - co wykazała już wstępna analiza obwodów zawartych na płytce odbiornika, jak i na podstawie informacji zawartych w dokumentacji firmy *Te-*

*lecontrolli*). Na schemacie widoczne są dwa typy filtrów, których parametry są istotne ze względu na wybór sposobu kodowania przesyłanych danych. Pierwszy z filtrów jest filtrem dolnoprzepustowym, który ogranicza szybkość przesyłania danych, jak też w sposób istotny wpływa na zasięg transmisji. Jak łatwo zauważyć, występuje tu pewien kompromis pomiędzy zasięgiem a szybkością przesyłanych danych. Ogólnie mówiąc, przesyłając dane wolniej (węższe pasmo filtru DP), można uzyskać większy zasięg przy stałej mocy nadajnika (mniejsze pasmo przepustowe filtru -

List. 4. Funkcja dekodowania danych z odbiornika radiowego

```
void int_ext0(void) interrupt 0 using 2
{
    unsigned int tmp;
    char p_th,state,cntbyte;
    bit p_in,c_in;

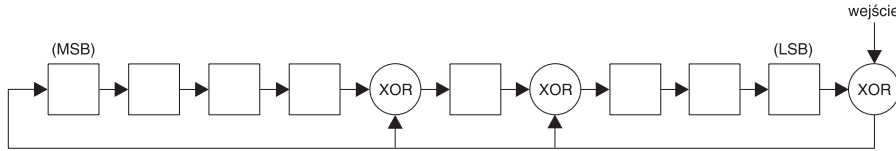
    TL0=TL1=0; TH0=TH1=0; TR1=1; ET1=0;
    tmp=0; cntbit=1; state=0; cntbyte=0;

    for(;;)
    {
        p_in=in;
        while(p_in==in) if (TH0>8) return;

        //Poniżej kod odpowiedzialny za ignorowanie krótkich impulsów
        p_th=TH0; TL0=0; TH0=0;
        while(p_in!=in) if(TH0>1) break;
        if(p_in==in) { TL0=TL1; TH0=TH1; continue; }
        else { TL1=TL0; TH1=TH0; }

        c_in=in;

        switch(state)
        {
            case 0:
                if(!p_in&&c_in&&p_th>2&&p_th<5) state=1;
                else
                    if(!p_in&&c_in&&p_th>4&&p_th<8)
                        { state=2; tmp<=1; cntbit++; }
                    else return;
                break;
            case 1:
                if(p_in&&!c_in&&p_th>2&&p_th<5)
                    { state=0; tmp<=1; tmp|=1; cntbit++; }
                else return;
                break;
            case 2:
                if(p_in&&!c_in&&p_th>4&&p_th<8)
                    { state=0; tmp<=1; tmp|=1; cntbit++; }
                else
                    if(p_in&&!c_in&&p_th>2&&p_th<5) state=3;
                    else return;
                break;
            case 3:
                if(!p_in&&c_in&&p_th>2&&p_th<5)
                    { state=2; tmp<=1; cntbit++; }
                else return;
                break;
        }
        if(cntbit>=8) { cntbit=0; buf[cntbyte]=tmp; tmp=0; cntbyte++; }
        if (cntbyte>=MAXBYTE)
        {
            TR1=0; ET1=1; EX0=1; rdy=1;
            return;
        }
    }
}
```



Rys. 7. Algorytm wyliczania CRC

większa wartość współczynnika sygnał-szum - możliwość odbioru słabszych sygnałów) i na odwrót. Drugi z filtrów, przez który przechodzi sygnał, jest filtrem górnoprzepustowym utworzonym przez kondensator sprzęgający (sprzężenie zmiennoprądowe).

Aby informacja mogła być bez przeszkód przesłana przez te dwa filtry, bardzo istotny jest właściwy wybór sposobu kodowania danych. Należy zwrócić uwagę na minimalną szerokość impulsu (lub przerwy między impulsami), która musi być dostosowana do szerokości pasma filtru dolnoprzepustowego, jak również na maksymalną szerokość impulsu (przerwy), która z kolei musi pozostawać w odpowiedniej relacji do stałej czasowej wyznaczonej przez kondensator sprzęgający wraz z pozostałą częścią obwodu. Ze względu na zmiennoprądowe sprzężenie sygnału bardzo istotne jest, aby sposób kodowania danych zapewniał właściwe zrównoważenie składowej stałej. Czyli najlepiej aby kodowany sygnał miał wartość logiczną „1” przez 50% czasu i wartość „0” przez pozostałe 50% czasu. Zrównoważenie składowej stałej może być osiągnięte na wiele sposobów. Jednym z najpopularniejszych jest wykorzystanie kodu Manchester, w którym bity są kodowane bifazowo - jedynie logicznej odpowiada zmiana poziomu sygnału w połowie taktu z wysokiego na niski, zaś

zeru logicznemu - zmiana poziomu sygnału w połowie taktu z niskiego na wysoki (a więc analogicznie jak dla kodowania w standardzie RC5). Z zachowaniem równowagi składowej stałej wiąże się potrzeba ograniczenia liczby kodowanych bitów jako „1” (lub „0”), występujących tuż po sobie. Kod Manchester również tutaj sprawdza się doskonale (choć jego wadą jest podwojenie liczby impulsów potrzebnych do przesłania danego komunikatu).

Innym ważnym problemem jest właściwy dobór wartości progowej napięcia dla układu formowania danych (komparatora). Ustawienie wartości tego napięcia (źródło napięciowe na rysunku 5) większej od zera powoduje aktywowanie blokady szumów. Zbyt mała wartość tego napięcia sprawia, że szumy o odpowiednio dużej amplitudzie przenoszą się na wyjście w postaci krótkich, szpilkowych impulsów. Wówczas prawidłowe odtworzenie sygnału użytecznego wymaga specjalnych zabiegów ze strony oprogramowania. Z kolei zbyt duża wartość tego napięcia powoduje zmniejszenie czułości odbiornika również dla sygnału użytecznego. W zastosowanym odbiorniku hybrydowym napięcie to jest ustalone przez producenta, zwalniając tym samym użytkownika od konieczności jego doboru.

Biorąc pod uwagę powyższe założeń, wybrano sposób kodowania danych identyczny jak w przypadku standardu RC5. Zachowano też identyczną szybkość przesyłanych danych. Ponieważ nie dysponowano szczegółowymi danymi odnośnie pasma przenoszenia filtru dolnoprzepustowego odbiornika RR3 ani wartością stałej czasowej filtru górnoprzepustowego, zatem oszacowania minimalnej i maksymalnej szerokości impulsu były dość trudne do przeprowadzenia. Znana była jedynie graniczna szybkość odbieranych danych wynosząca dla odbiornika RR3 - 2 kHz. Przyjęto więc za zadowalające parametry transmisji danych

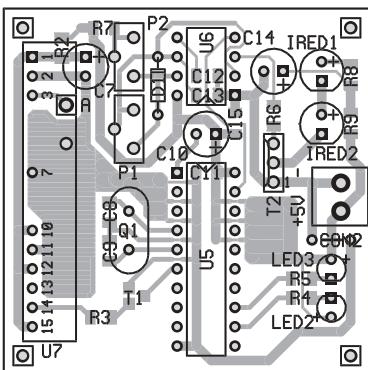
```

List. 5. Funkcja obliczająca CRC
void crc_update(unsigned char x)
{
    unsigned char tmp,f,i;
    f=1;
    for(i=0;i<8;i++)
    {
        tmp=CRC; CRC=(CRC>>1)&0x73;
        CRC|=(((tmp&0x01)^((x&f)?0x01:0))?)
            (((tmp^0x18)>>1)|0x80:tmp>>1)&0x8c;
        f<<=1;
    }
}
    
```

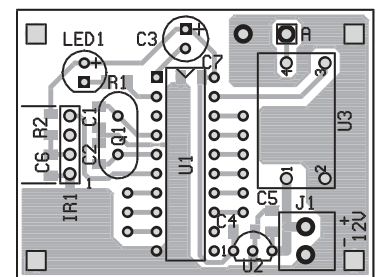
charakterystyczne dla standardu RC5: minimalna szerokość impulsu (przerwy) równa 890µs, co daje maksymalną częstotliwość około 561 Hz, a więc znacznie poniżej częstotliwości granicznej.

Na list. 4 zamieszczono kod funkcji w języku C dekodującej dane z toru odbiornika radiowego. Kod jest analogiczny jak w przypadku funkcji dekodowania danych w standardzie RC5 (wykorzystano ten sam algorytm oparty na grafie przejść z rys. 4). Funkcja odbiera liczbę bajtów określoną przez stałą symboliczną MAXBYTE, umieszczając je kolejno w tablicy znakowej o nazwie *buf*. W funkcji zawarto fragmenty kodu, dzięki którym są ignorowane w analizowanym sygnale z odbiornika radiowego krótkie impulsy (poniżej 512µs), będące wynikiem interferencji radiowych lub szumów (wykorzystano tutaj obydwaj liczniki sprzętowe T0 i T1 pracujące współbieżnie). Użycie funkcji we własnym programie jest analogiczne jak opisano to przy okazji dekodowania danych w standardzie RC5. Również funkcja, która pozwala zakodować dane dla nadajnika radiowego, jest identyczna jak w przypadku standardu RC5, stąd też nie ma potrzeby zamieszczania jej w całości.

Długość całego transmitowanego pakietu danych z przemiennika IR-RF do przemiennika RF-IR wynosi 32 bity (4 bajty). Pierwsze 8 bitów stanowi preambułę (sekwencja rozbiegowa zawsze równa 11001010<sub>(2)</sub>).



Rys. 8. Rozmieszczenie elementów na płycie konwertera radio->RC5



Rys. 9. Rozmieszczenie elementów na płycie konwertera RC5->radio

Kolejne 16 bitów, z których istotne jest tylko 14 bitów mniej znaczących, stanowi kompletny kod polecenia RC5. Ostatnie 8 bitów jest sumą kontrolną CRC. Według [9] zaleca się stosowanie 16-bitowej sumy kontrolnej opartej na standardzie X.25 (ISO3309), jednak ponieważ w rozważanym przypadku pakiet danych miał bardzo niewielką długość, dlatego przyjęto za wystarczające zastosowanie 8-bitowego CRC, liczonego według wielomianu  $x^8+x^5+x^4+1$ , typowego dla układów pracujących z jedнопроводową magistralą firmy *Dallas/Maxim*.

Na rys. 7 pokazano sposób wyliczania wielomianu CRC, zaczerpnięty wprost z dokumentacji firmy *Dallas/Maxim*. Na list. 5 przedstawiono jedną z wielu możliwych implementacji tego algorytmu w języku C, zastosowaną w projekcie.

Funkcja wykorzystuje zewnętrzną (globalną) zmienną znakową CRC, przechowującą bieżącą wartość wielomianu. Funkcja aktualizuje CRC od razu dla całego bajtu (8-bitowy argument funkcji).

Radiowy przedłużacz pilotów można w sposób bardzo prosty, poprzez niewielką modyfikację oprogramowania, rozbudować np. o funkcję akceptującą na wejściu dowolnego pilota RC5 (o dowolnym lub wcześniej ustalonym adresie urządzenia RC5), jak również możliwa jest opcja zamiany kodów rozkazów, adresów urządzenia RC5 itd.

**Zbigniew Hajduk**

Literatura:

- [1] Knowledge Base, Philips RC-5 Protocol, <http://www.xs4all.nl/~sbp/knowledge/ir/rc5.htm>
- [2] Biały R., Programowe dekodowanie sygnałów zdalnego sterowania, *Elektronika Praktyczna* 12/2002, strony 95-98.
- [3] An Efficient Algorithm for Decoding RC5 Remote Control Signals, <http://www.clearwater.com.au/pcm-9574/rc5/rc5.html>
- [4] Putzger J., Decoding IR Remote Controls, [http://www.ee.washington.edu/circuit\\_archive/text/ir\\_decode.txt](http://www.ee.washington.edu/circuit_archive/text/ir_decode.txt)

- [5] RT4 Radio Transmitter Module with SAW Resonator and External Antenna, *Telecontrolli Datasheet*, <http://www.telecontrolli.com>
- [6] RR3 Super Regenerative Radio Receiver With Laser Trimmed Inductor, *Telecontrolli Datasheet*, <http://www.telecontrolli.com>
- [7] Smith K., Antennas for Low Power Applications, RFM Application Note AN36.
- [8] Evjen P. M., SRD Antennas, Chipcon Application Note AN003.
- [9] ASH Transceiver Designers Guide, RFM Application Note.
- [10] ASH Transceiver Software Designer's Guide, RFM Application Note.
- [11] Virtual Wire Development Kit Manual for DR1300-DK, RFM Application Note.

*Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: **pcb.ep.com.pl** oraz na płycie CD-EP4/2004B w katalogu PCB.*