

# 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 unieumożliwia 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



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)

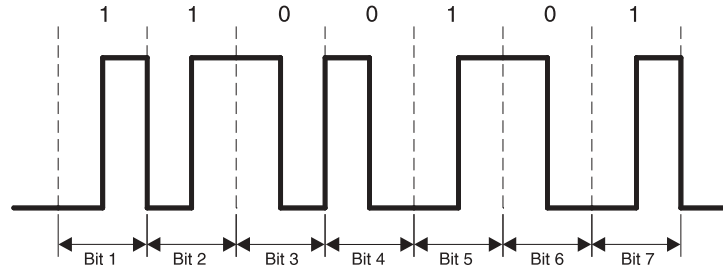
Część sprzętowa radiowego przedłużacza pilotów, zawierająca typowe komponenty, jest jak widać bardzo prosta. Właściwe cechy funkcjonalne nadaje przedłużaczowi oprogramowanie. Przygotowano je w języku C i skompilowano kompilatorem firmy KEIL.

### 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:

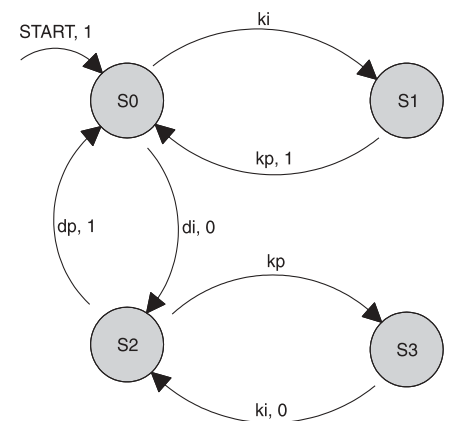
- ki - krótki impuls,
- kp - krótka przerwa,
- di - długi impuls,
- dp - długa przerwa,

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