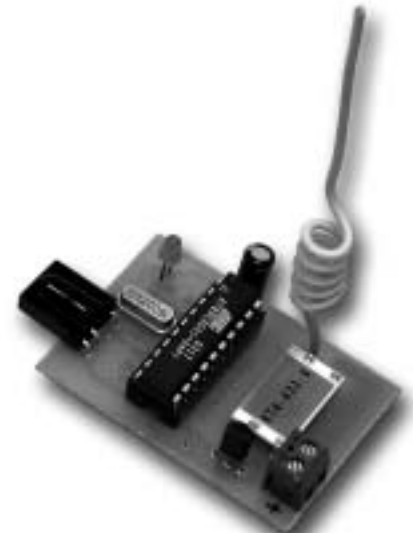


Radiowy przedłużacz pilotów, część 2

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 przeznaczonych 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;
```

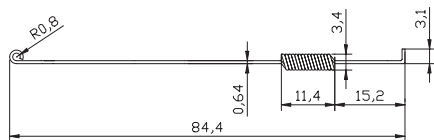
```
// while (TR1);
```

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).

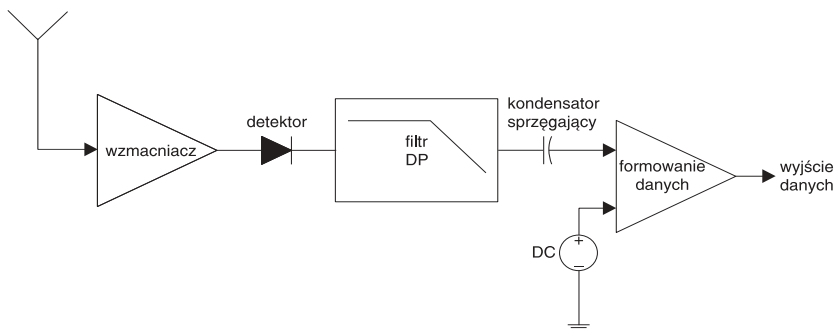
Dla obu pokazanych wyżej listingów zmienne bitowe *in* oraz *out* odnoszą się bezpośrednio do odpowiednich portów mikrokontrolera i ich definicja dla schematów z rysunków 1 i 2 powinna być następująca:

```
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 średnicy 2,6mm, powinna wynosić 11,4mm. Cała antena winna mieć długość około 84mm [11]. Jako materiał na antenę należy zastosować izolowany przewód (najlepiej srebrzanekę) 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;

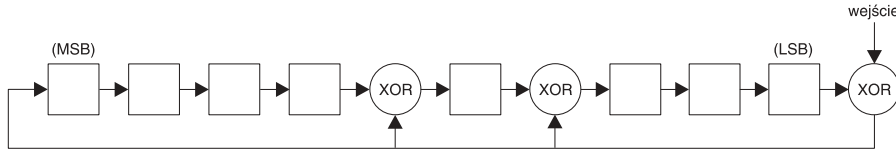
    TL0=TL1=0; TH0=TH1=0; TR1=1; ET1=0;
    tmp=0; cntbit=1; state=0; cntbyte=0;

    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 szумы 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 zależności, 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

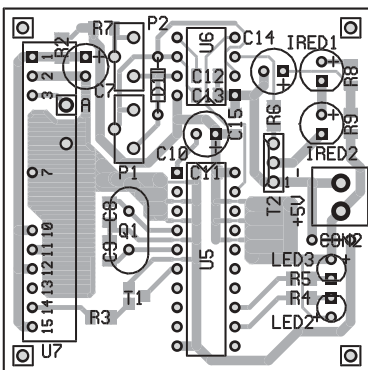
```

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;
    }
}
    
```

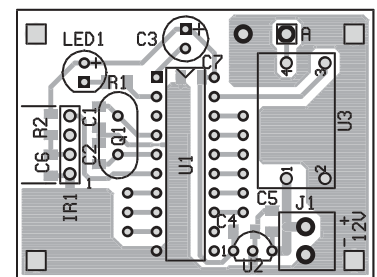
charakterystyczne dla standardu RC5: minimalna szerokość impulsu (przerwy) równa 890µs, co daje maksymalną częstotliwość około 561 Hz, a więc znacznie poniżej częstotliwości granicznej.

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₍₂₎).



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

- [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**.*