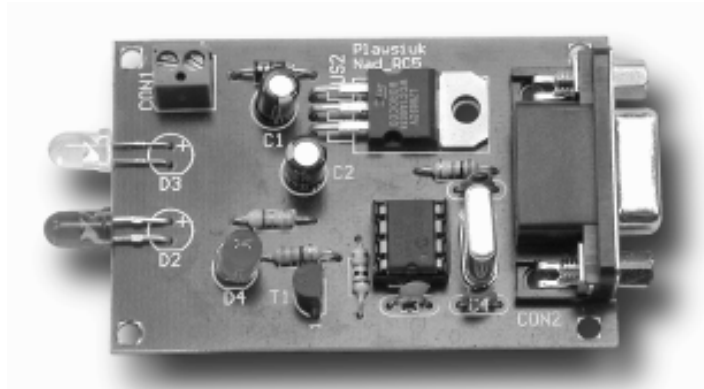


Uniwersalny nadajnik RC5/SIRC, część 1

AVT-5104

Nadajnik zdalnego sterowania kojarzy nam się zazwyczaj z pudełkiem z wieloma klawiszami. Opisy takich nadajników były niejednokrotnie publikowane na łamach EP. Nadajnik prezentowany w artykule ma nieco odmienną budowę, gdyż nie posiada ani jednego przycisku, a może wygenerować dowolny kod w standardzie RC5 oraz SIRC.

Rekomendacje: przyrząd szczególnie przydatny do prac serwisowych przy naprawie odbiorników telewizyjnych, magnetowidów, odtwarzaczy DVD i sprzętu audio. Przyda się także wszystkim fanom eksperymentów ze zdalnym sterowaniem w dwóch najbardziej popularnych standardach kodowania.



Nadajnik współpracuje z komputerem poprzez port szeregowy RS232. Dzięki takiemu rozwiązaniu ma niewielkie wymiary, a umożliwia wysłanie dowolnej komendy za pomocą klawiatury komputera lub odpowiedniego oprogramowania. Dzięki temu jest możliwe sterowanie urządzeniami znajdującymi się w innych pomieszczeniach, ponieważ komunikacja pomiędzy komputerem a nadajnikiem odbywa się poprzez interfejs RS232, co pozwala na oddalenie nadajnika od komputera nawet o kilkanaście metrów. Wysłanie komendy odbywa się za pomocą znaków ASCII, więc obsługa nadajnika może być wykonywana przez dowolny program terminalowy - na przykład HyperTerminal - w który wyposażony jest każdy system Windows.

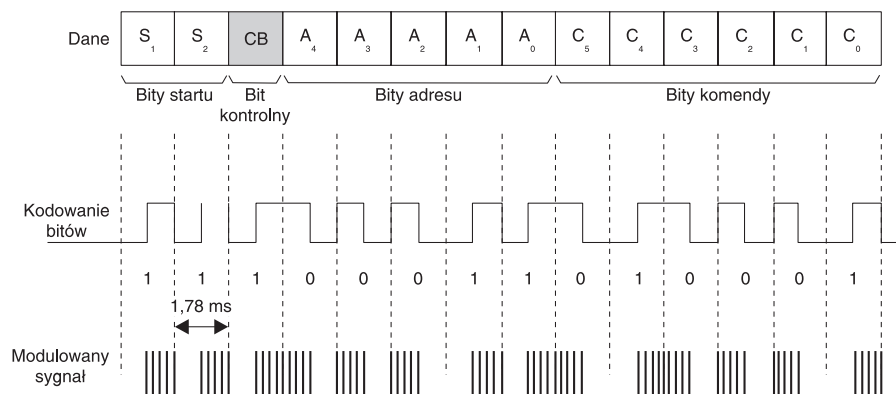
Obsługa dwóch systemów kodowania informacji umożliwia sterowanie pracą większości urządzeń elektronicznych. Bardzo popularny system RC5 jest stosowany przez wiele firm, między innymi przez firmę Philips, a system SIRC głównie przez firmę Sony. W systemie RC5 sygnały nadawane są z sygnałem nośnym o częstotliwości 36 kHz, a w systemie SIRC częstotliwość ta wynosi 40 kHz. Zastosowanie w nadajniku mikrokontrolera umożliwia łatwe dobranie tej częstotliwości w zależności od systemu, w którym jest wysyłana komenda. Przełączanie pomiędzy systemami

dokonyje się poprzez naciśnięcie jednego klawisza. Jeśli kolejne komendy będą wysyłane w tym samym systemie, to nie trzeba ponownie wybierać systemu. Również powtórzenie tej samej komendy nie wymaga ponownego jej wpisywania, gdyż znajduje się w pamięci i wystarczy tylko ponowne jej zatwierdzenie.

Ponieważ sposób transmisji danych w obydwu systemach jest różny, więc dla obu systemów zostały opracowane odmiennie procedury. Sposób kodowania danych dla systemu RC5 przedstawiono na rys. 1.

Tajniki RC5

W tym systemie każdy rozkaz składa się z 14 bitów. Rozpoczęcie transmisji następuje po dwóch bitach startu (S1, S2) - obydwa bity mają wartość „1”. Następnie wysyłany jest bit kontrolny (CB). Bit kontrolny pozwala określić, czy naciśnięty przycisk na pilocie jest przytrzymywany, czy został naciśnięty ponownie. Gdy przycisk jest przytrzymywany, to wysyłane są kolejne sygnały i bit ten ma taką samą wartość. Jeśli przycisk został zwolniony i ponownie naciśnięty, to bit kontrolny zmieni stan na przeciwny. Następnie jest wysyłany adres urządzenia, do którego jest skierowana komenda. Adres ten składa się z pięciu bitów, co pozwala wybrać 32 urządzenia. W następnym etapie wysyłanych jest 6



Rys. 1. Przebiegi czasowe sygnałów podczas wysyłania rozkazu w systemie RC5

bitów komendy, co umożliwia wysłanie 64 różnych poleceń. Każdy zakodowany bit w tym systemie trwa 1,776 ms i składa się z dwóch części - 888 μs przerwy (poziom niski) i 888 μs impulsu (poziom wysoki). Czy wysłany bit jest jedyneką, czy zerem określa pierwsza część kodowanego bitu - jest zerem lub jedyneką. Jeśli ta część bitu jest jedyneką, to poprzez diody nadawcze emitowany jest sygnał o częstotliwości 36 kHz, jeżeli część bitu jest zerem, to żaden sygnał nie jest emitowany.

Aby wysłać bit, interpretowany przez odbiornik jako „1”, należy przez pierwsze 888 μs wyłączyć diodę nadawczą, a przez 888 μsysterować ją sygnałem o częstotliwości 36 kHz. W przypadku wysyłania logicznego „0”, jest odwrotnie, przez pierwsze 888 μs należy sterować diodę nadawczą sygnałem o częstotliwości 36kHz, a przez następne 888 μs pozostawić diodę wyłączoną. W ten sposób należy wysłać wszystkie bity transmitowanego rozkazu.

Tajniki SIRC

Sposób kodowania w systemie SIRC jest przedstawiony na rys. 2. Każdy rozkaz składa się z jed-

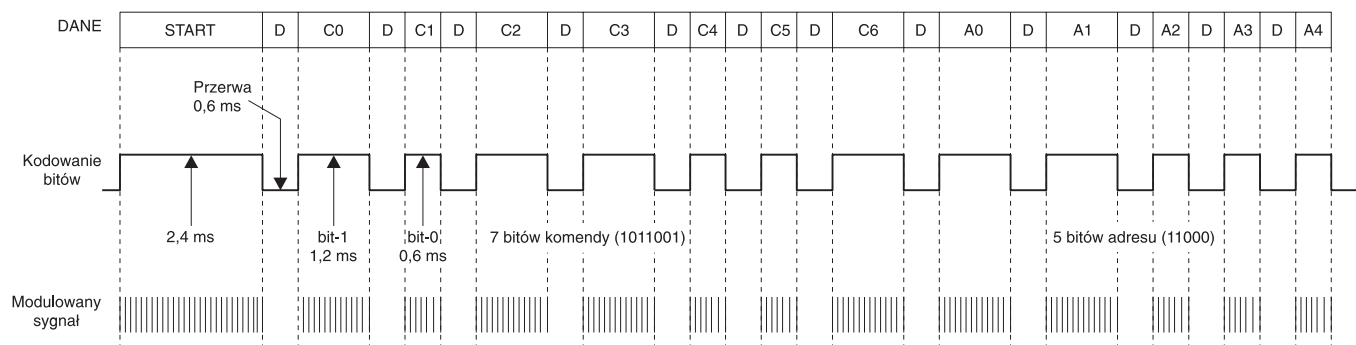
nego bitu startu i 12 bitów danych. Można wybrać 32 urządzenia i 128 komend. Po wysłaniu bitu startu następuje przerwa o długości 0,6 ms, następnie wysyłane jest siedem bitów komendy, poczynając od bitu najmniej znaczącego. Po wysłaniu komendy wysyłane jest pięć bitów adresu urządzenia. Sposób kodowania stanów logicznych polega na tym, że dla zera logicznego transmitowany jest przebieg o częstotliwości 40 kHz przez czas 0,6 ms, a dla jedynki logicznej przez czas 1,2 ms. Dodatkowo, pomiędzy kolejnymi bitami następuje przerwa w transmisji o czasie 0,6 ms. Ze względu na zróżnicowany czas trwania stanu zera logicznego i jedynki, wysłanie całego pakietu danych nie zajmuje za każdym razem tyle samo czasu. Jest on zależny od wartości poszczególnych bitów adresu urządzenia i komendy.

Budowa i działanie nadajnika

Schemat elektryczny nadajnika przedstawiono na rys. 3. Składa się on zaledwie z kilku elementów, gdyż wszystkie niezbędne funkcje pełni układ US1. Układ ten jest mikroprocesorem z we-

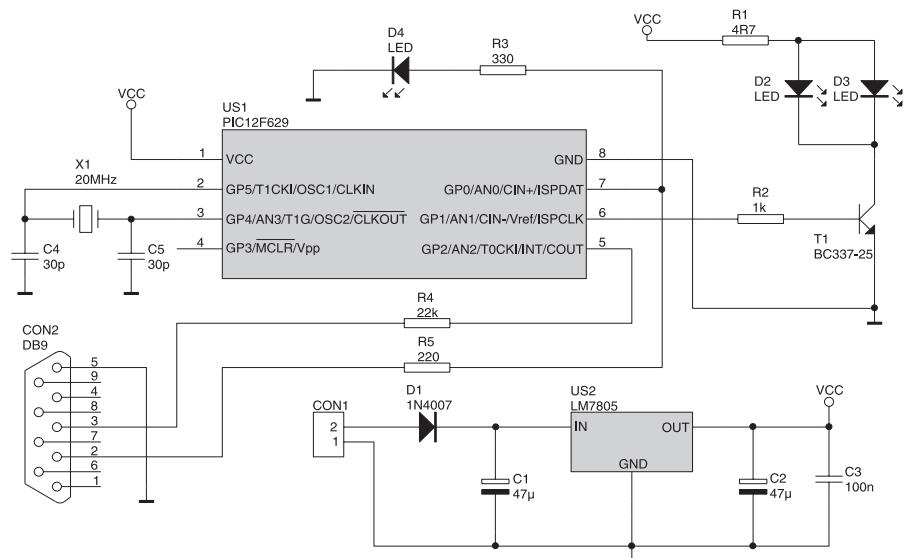
wewnętrzną pamięcią Flash o pojemności 1k x 14 b. Dzięki zastosowaniu mikrokontrolera, wszystkie operacje związane z odbieraniem i wysyłaniem danych są wykonywane programowo, w związku z czym nie ma potrzeby stosowania dodatkowych układów. Do procesora dołączono zewnętrzny rezonator kwarcowy o częstotliwości 20 MHz, wskutek czego cykl maszynowy wynosi tylko 200 ns. Pozwala to na bardzo dokładne ustalenie częstotliwości generowanych sygnałów nośnych potrzebnych do wysyłania sygnałów zdalnego sterowania.

Wejście procesora przeznaczone do zerowania (!MCLR) nie zostało wykorzystane, ponieważ sygnał zerowania po włączeniu zasilania jest generowany wewnątrz mikrokontrolera. Jako nadajniki promieniowania podczerwonego pracują diody D2 i D3. Diody te są sterowane przez procesor za pomocą wzmacniacza zbudowanego z tranzystora T1. Rezystor R1 ogranicza prąd płynący przez diody, a R2 prąd bazy tranzystora T1. Do sygnalizacji pracy nadajnika zastosowano diodę D4. Po każdym wysłaniu komendy dioda błysnie. Równocześnie sygnał zasilający diodę D4 jest doprowadzony, poprzez rezystor R5, do złącza CON2. Połączenie to sprawia, że po wysłaniu wybranej komendy, jej numer zostanie wysłany z powrotem do komputera i będzie można sprawdzić, czy komenda została wpisana prawidłowo. Komunikacja z komputerem odbywa się z prędkością 9600 bd. Wysyłane dane do komputera nie są dopasowane napięciowo do norm standardu RS232, gdyż stanowi „0” odpowiednia 0 V, a stanowi „1” napięcie 5 V. Układy obsługujące porty szeregowo



Rys. 2. Przebiegi czasowe podczas wysyłania rozkazu w systemie SIRC

w komputerach rozpoznają prawidłowo tak określone stany logiczne. Sygnały z komputera do mikrokontrolera również nie są dopasowane napięciowo, ale do zabezpieczenia procesora wystarczy ograniczyć prąd wejściowy przez rezystor szeregowy R4. Na wyjściu portu szeregowego komputera występują napięcia +12 V lub -12 V. Zastosowanie rezystora R4 w połączeniu z wewnętrznymi diodami zabezpieczającymi, zawartymi wewnątrz procesora, powoduje, że napięcie większe od $VCC+0,4V$ jest zwierane do plusa, a napięcie mniejsze od $GND-0,4V$ jest zwierane do masy, w konsekwencji na wejściu procesora występują bezpieczne wartości napięć: -0,4V lub 5,4V. Takie sprzężenie nadajnika z portem szeregowym znacznie uprościło jego budowę. Do stabilizowania napięcia zasilającego zastosowano układ US2, który wraz z kondensatorami C1...C3 dostarcza odfiltrowanego napięcia o wartości 5V.



Rys. 3. Schemat elektryczny uniwersalnego nadajnika RC5/SIRC

Transmisja danych w podczewieni polega na odebraniu przez procesor cyfr kodu, przetworzeniu tych cyfr na kod systemu RC5 lub SIRC i odpowiednimysterowaniem tranzystora sterującego diodami nadawczymi. Po wysłaniu

podanego kodu do komputera zostaje wysłana zwrotna informacja, jaki kod został wyemitowany.

Jak wcześniej wspomniano, obydwa systemy kodowania informacji różnią się między sobą. Do ich obsługi trzeba więc zastosować różne procedury.

Na list. 1 przedstawiono procedury wysyłania danych w systemie RC5, przy zastosowaniu rezonatora kwarcowego o częstotliwości 20 MHz. Główną procedurą jest procedura *RC5_send(int address, int command)*. Przykładowo, aby wysłać do urządzenia o numerze 10 komendę o numerze 36, należy wywołać tę procedurę z parametrem: *RC5_send(10,36)*, w wyniku czego zostaną wysłane wszystkie bity rozkazu. Procedura ta dołącza bity startu i bit kontrolny, tworząc pełną ramkę danych zgodną z wymaganiami RC5.

Wartość parametru adresu urządzenia musi zawierać się w przedziale 0...31, a parametr komendy w 0...63. Procedura *RC5_send* sprawdza wszystkie bity otrzymanych parametrów i w zależności od tego, czy sprawdzany bit jest jedynka, czy zerem, zostaje wykonana procedura *stan_1()* lub *stan_0()*. Sprawdzenie stanu bitu polega na przesunięciu w lewo całego bajtu, wtedy najstarszy bit znajdzie się w rejestrze *STATUS* jako bit *C* i dopiero wtedy można sprawdzić stan tego bitu. Taka operacja jest wykonywana dla wszystkich bitów adresu urządzenia lub komendy. W systemie RC5 najpierw jest wysyłany adres urzą-

List. 1. Procedury służące do wysłania dowolnego kodu w systemie RC5

```

//*****
// wysyla logiczne zero f=36kHz kwarc 20MHz //
//*****
stan_0()
{int i;
 led_ir=0; //wylacz diode
 for(i=32;i>0;i-) //powtorz 32 razy
 {led_ir=1;
 delay_us(13); //stan wysoki 13us
 delay_cycles(4); //+800ns=13,8us
 led_ir=0;
 delay_us(13); //stan niski 13us
 delay_cycles(1); //+200ns=13,2us+3cykle=13,8us
 } //13,2+(3*200)=13,8us
 delay_us(875); //875+13=888us
}

//*****
// wysyla logiczna jedynke f=36kHz kwarc 20MHz //
//*****
stan_1()
{int i;
 led_ir=0;
 delay_us(888);

 for(i=32;i>0;i-) //powtorz 32 razy
 {led_ir=1; //zapal diode
 delay_us(13); //czekaj 13us
 delay_cycles(4); //+800ns=13,8us
 led_ir=0; //zgas diode
 delay_us(13); //stan niski 13us+3cykle przez licznik
 delay_cycles(1); //+200ns+600ns=13,8us
 } //razem 2*13,8=27,6us=36,2khz
}

//*****
// Wysyla podana sekwencje w rc5
//
// toggle=1 //
// address 0-31 //
// command 0-63 //
//*****
RC5_send(int address, int command)
{ int i;
 address|=0b11100000;
 command<<=2;
 for(ii=0;ii<=7;ii++) //wysyla address+2bity startu =1,
 { //+ togglebit=1
 address<<=1; //przesun w lewo o jeden
 if(bit_test(status,0)) stan_1(); //jesli c=1 to stan_1()
 else stan_0(); //jesli 0 to stan_0()
 }
 for(ii=0;ii<=5;ii++) //wysyla 5bitow command
 {
 command<<=1; //przesun w lewo o jeden
 if(bit_test(status,0)) stan_1(); //jesli c=1 to stan_1()
 else stan_0(); //jesli 0 to stan_0()
 }
}
//*****
    
```

dzenia poczynając od najstarszego bitu, a następnie numer komendy również od bitu najstarszego. Procedury *stan_0()* i *stan_1()* odpowiednio sterują diodami nadawczymi, zgodnie z wymaganiami standardu - czas trwania bitu 1,788 ms, częstotliwość 36 kHz. Wysłanie jedynek logicznych (procedura *stan_1()*) powoduje wyłączenie diody nadawczej na 888 µs, a następnie przez kolejne 888 µs dioda jest sterowana sygnałem o częstotliwości 36 kHz. Wysłanie zera logicznego (procedura *stan_0()*) powoduje natomiast przez pierwsze 888 µs sterowanie diody sygnałem o częstotliwości 36kHz, a przez kolejne 888 µs dioda jest wyłączona.

Do transmisji w systemie SIRC stosowane są procedury przedstawione na list. 2. Główną procedurą jest procedura *Sony_send(int address, int command)*. Podobnie jak w przypadku wysyłania danych w systemie RC5, procedurę należy wywołać z parametrami określającym adres urządzenia i komendę. Dla standardu SIRC adres urządzenia może zawierać się w przedziale 0...31, a komenda w 0...127. Po wywołaniu tej procedury następuje wysłanie bitu startu, a następnie komendy i adresu urządzenia. Procedura wysyła pełną ramkę danych wraz z przerwami pomiędzy kolejnymi bitami. W systemie SIRC w pierwszej kolejności wysyłany jest

WYKAZ ELEMENTÓW

Rezystory

R1: 4,7Ω
R2: 1kΩ
R3: 330Ω
R4: 22kΩ
R5: 220Ω

Kondensatory

C1, C2: 47µF/16V
C3: 100nF
C4, C5: 30pF

Półprzewodniki

D1: 1N4007
D2, D3: dowolna dioda nadawcza w podczerwieni
D3: LED 5mm czerwona
T1: BC337-25
US1: PIC12F629 zaprogramowany
US2: LM7805

Różne

CON1: ARK2(3,5mm)
CON2: DB9 żeńskie
X1: rezonator kwarcowy 20MHz
Podstawka DIP8

numer komendy, poczynając od najmłodszego bitu, a następnie adres urządzenia, również od najmłodszego bitu. Podobnie jak w przypadku systemu RC5, sprawdzone są wszystkie bity adresu i komendy, i w zależności od ich wartości wykonywane są procedury: *sony_stan_1()* - dla jedynek logicznych i *sony_stan_0()* - dla zera logicznego. Kodowanie jedynek logicznych polega na wysyłaniu przebiegu o częstotliwości 40 kHz przez czas 1,2 ms. Realizuje to procedura *sony_stan_1()*. Ponieważ okres sygnału o częstotliwości 40 kHz wynosi 25 µs, to do odliczenia czasu 1,2 ms należy wykonać 48 błysnięć diodą nadawczą. Podczas wysyłania logicznego zera dioda musi błyskać przez 0,6 ms. Realizuje to procedura *sony_stan_0()* - w czasie 0,6 ms dioda błysnie 24 razy. Pomiedzy kolejnymi bitami musi być przerwa w transmisji wynosząca 0,6 ms. Jest ona realizowana przez procedurę *sony_delay()*.

Krzysztof Piłwsiuk, AVT
krzysztof.piłwsiuk@ep.com.pl

Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/marzec03.htm> oraz na płycie CD-EP3/2003B w katalogu PCB.

List. 2. Procedury służące do wysyłania danych w systemie SIRC

```
//*****
//wysyla logiczne zero f=40kHz,kwarc 20MHz //
//stan1 nadawanie przez 0,6ms //
//dla f=40kHz cykl=25uS, dla 0,6ms trzeba powtorzyc 24razy //
//*****
sony_stan_0()
(int i;

    led_ir=0; //zgas diode
    for(i=24;i>0;i--) //powtorz 24razy
    {led_ir=1; //zapal diode
      delay_us(12); //zapal na 12us
      delay_cycles(2); //+400ns =12,4us
      led_ir=0; //zgas diode
      delay_us(12); //zgas na 12us +600ns=12,6us
    }
}
//*****
//*****
//wysyla logiczna jedynke f=40kHz,kwarc 20MHz //
//stan1 nadawanie przez 1,2ms //
//dla f=40kHz cykl=25uS, dla 1,2ms trzeba powtorzyc 48razy //
//*****
sony_stan_1()
(int i;
    led_ir=0; //zgas diode
    for(i=48;i>0;i--) //powtorz 48 razy
    {led_ir=1; //zapal diode
      delay_us(12); //na 12us
      delay_cycles(2); //+400ns=12,4ns
      led_ir=0; //zgas diode
      delay_us(12); //na 12us+3cykle=12,6us
    }
}
//*****
//*****
//opoznienie pomiedzy bitami 600us //
//*****
sony_delay()
{
    led_ir=0;
    delay_us(599);
}
//*****
//*****
// wysyla kod podany jako parametr //
//address-adres urzadzenia 0-31, command-komenda 0-127 //
// //
//*****
sony_send(int address, int command)
{ int ii;
  sony_stan_1(); //bit startu I \_2,4ms
  sony_stan_1(); //bit startu II/
  sony_delay(); //delay 0,6ms
  for(ii=0;ii<=6;ii++) //wysyla 7 bitow command
  {
    command>>=1; //przesun w prawo
    if(bit_test(status,0)) sony_stan_1(); //jeśli 1 to sony_stan_1()
    else sony_stan_0(); //jeśli 0 to sony_stan_0()
    sony_delay(); //przerwa miedzy bitami
  }

  for(ii=0;ii<=4;ii++) //wysyla 5 bitow adresu
  {
    address>>=1; //przesun w prawo
    if(bit_test(status,0)) sony_stan_1(); //jeśli 1 to sony_stan_1()
    else sony_stan_0(); //jeśli 0 to sony_stan_0()
    sony_delay(); //przerwa miedzy bitami
  }
}
//*****
```