

# radioRelay

## System sterowania bezprzewodowego

*Jak łatwo zauważyć, zdalnie sterowane, bezprzewodowe moduły przekazników cieszą się niesłabnącą popularnością i stanowią temat wielu ciekawych projektów. W sumie nie powinno to dziwić, ponieważ urządzenia tego typu niejednokrotnie znacznie ułatwiają codzienną egzystencję lub zwyczajnie udostępniają nam nowe możliwości.*

**Rekomendacje:** moduł przyda się do wykonania zdalnego sterowania np. w inteligentnym budynku.

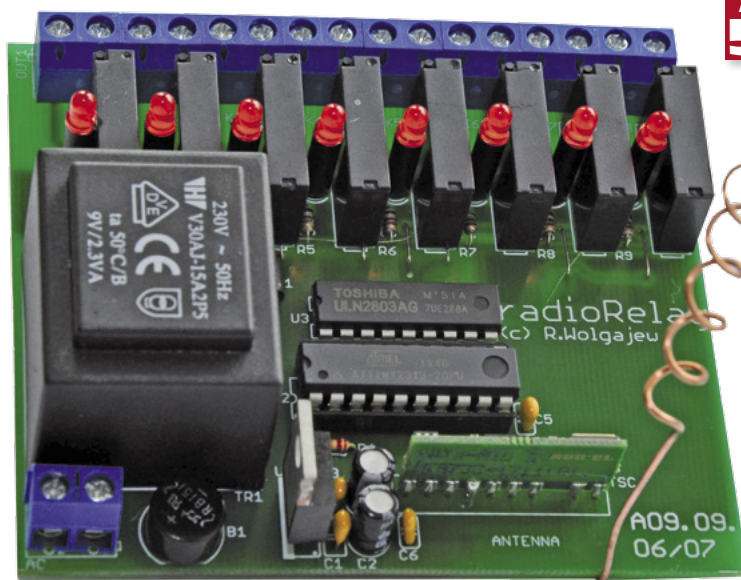
Projekty jak i urządzenia, z którymi się spotkałem „wertując” dziesiątki stron o tematyce elektronicznej nie wymagały do pracy nieprzerwanego dostępu do mediów, o które nie wszędzie łatwo (np. Internet lub zasięg sieci komórkowej). Biorąc to wszystko pod uwagę, postanowiłem zaprojektować i zbudować własne urządzenie o podobnej funkcjonalności, lecz pozbawione wad czy też wymagań, o których ogólnikowo wspominałem. Główne założenia projektowe, które przyświecały mi w trakcie wspomnianych prac były następujące:

- system ten ma zapewniać bezprzewodową, dwukierunkową transmisję danych na odległość (w terenie otwartym) do ok. 100 m,
- urządzenie sterujące (pilot) musi mieć niewielkie wymiary oraz długi czas pracy na zasilaniu bateryjnym,
- urządzenie sterujące (pilot) musi dostarczać informacji zwrotnej o wykonaniu lub odrzuceniu rozkazu sterującego (włączenia czy też wyłączenia przekaźnika),
- system ma być prosty i tani w budowie.

Tak oto powstał system radioRelay. Składa się on z Mastera (pilota jako urządzenie sterujące) i modułu Slave (moduł z przekaźnikami jako urządzenie wykonawcze).

### Nadajnik - Master

Schemat pilota sterującego, którego schemat pokazano na rysunku 1. Jest nieskomplikowany system mikroprocesorowy, którego sercem jest niewielki mikrokontroler ATtiny2313. Odpowiada on za obsługę pro-



**AVT  
5379**

stej klawiatury złożonej z 8 przycisków oraz transmisję (kodowanie/dekodowanie) danych. W torze radiowym pracuje transceiver RTX-MID-3V produkcji firmy Aurel. Wybór tego konkretnego układu transceivera nie był przypadkowy. Po pierwsze, moduł ten nie wymaga konfigurowania i innych czynności, poza obsługą wyprowadzeń sterujących. Moduł może również pracować (podobnie jak mikrokontroler sterujący) w specjalnym trybie PowerDown charakteryzującym się bardzo małym poborem prądu rzędu 8  $\mu$ A, co jest ważne dla urządzenia zasilanego bateryjnie. Nie bez znaczenia są również niewielkie wymiary wspomnianego modułu radiowego.

Na schemacie naszego pilota ważny Czytelnik zauważy zapewne nietypową konfigurację, jeśli chodzi o sposób włączenia przycisków, ponieważ, ich wspólne wyprowadzenia nie zostały (jak to zwykle bywa) dołączone do masy, tylko do wyprowadzenia INTO mikrokontrolera sterującego. Dziwne? Otóż, nie! Wykorzystano tutaj możliwość pracy tegoż układu w trybie PowerDown, czyli w trybie o bardzo niskim (nawet sporo mniejszym, niżeli transceiver) poborze mocy, z którego to możemy go wyprowadzić mikrokontroler właśnie dzięki wystąpieniu przerwania zewnętrznego INTO, zainicjowanego wciśnięciem któregośkolwiek przycisku sterującego. Zatem po uruchomieniu, program obsługi urządzenia ustawi port przycisków PORTB jako port wyjściowy oraz zeruje jego wyjścia, zaś pin INTO jako pin wejściowy z podciąganiem do napięcia zasilania. Następnie konfiguruje kilku innych bloków peryferyjnych i wprowadza mikro-

#### W ofercie AVT\*

AVT-5379 A  
AVT-5379 UK

AVT-5379 B

#### Podstawowe informacje:

Pilot systemu radioRelay:

- Zasilanie: bateria CR2450.
- Maksymalny prąd obciążenia (tryb uśpienia/nadawanie): 10  $\mu$ A/20 mA.
- Częstotliwość pracy transceivera: 433,92 MHz.
- Zasięg w terenie otwartym: ok. 100 m.
- Moduł wykonawczy systemu radioRelay:
- Napięcie zasilania: 230 V AC.
- Moc pobierana (tryb bezczynności/praca): poniżej 2 W.
- Liczba kanałów wykonawczych: 8.
- Maksymalny prąd styków: 5 A@250 V AC.
- Częstotliwość pracy transceivera: 433,92 MHz.
- Zasięg w terenie otwartym: ok. 100 m

#### Dodatkowe materiały na CD/FTP:

<ftp://ep.com.pl>, user: 32858, pass: 4285avne

- wzory płytek PCB
- karty katalogowe i noty aplikacyjne elementów oznaczonych w Wykazie elementów kolorem czerwonym

#### Projekty pokrewne na CD/FTP:

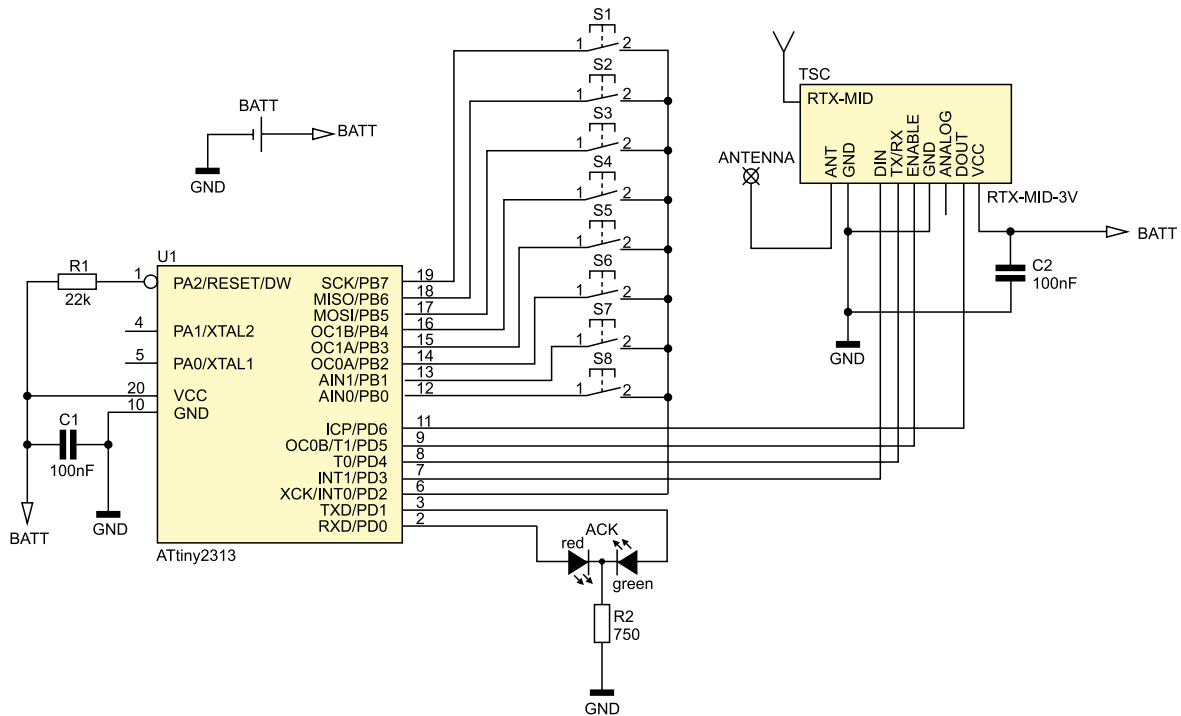
(wymienione artykuły są w całości dostępne na CD)  
AVT-5368 Programowalny moduł przekaźników (EP 11/2012)

#### \* Uwaga:

Zestawy AVT mogą występować w następujących wersjach:  
AVT xxxx UK to zaprogramowany układ. Tylko i wyłącznie. Bez elementów dodatkowych.  
AVT xxxx A płytka drukowana PCB (lub płytki drukowane, jeśli w opisie wyraźnie zaznaczono), bez elementów dodatkowych.  
AVT xxxx A+ płytka drukowana i zaprogramowany układ (czyli połączenie wersji A i wersji UK) bez elementów dodatkowych.  
AVT xxxx B płytka drukowana (lub płytki) oraz komplet elementów wymieniony w załączniku pdf  
AVT xxxx C to nic innego jak zmontowany zestaw B, czyli elementy wstawione w PCB. Należy mieć na uwadze, że o ile nie zaznaczono wyraźnie w opisie, zestaw ten nie ma obudowy ani elementów dodatkowych, które nie zostały wymienione w załączniku pdf  
AVT xxxx CD oprogramowanie (nieczęsto spotykana wersja, lecz jeśli występuje, to niezbędne oprogramowanie można ściągnąć, klikając w link umieszczony w opisie kitu)

Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! (UK, A, A+, B lub C). <http://sklep.avt.pl>

kontroler oraz moduł transceivera w tryb obniżonego ponoru energii. Od tej chwili każde naciśnięcie dowolnego przycisku powoduje wybudzenie procesora i realizację programu obsługi przerwania INTO, które jest odpowiedzialne za odczytanie stanu klawiszy, załączenie transceivera, wysłanie rozkazu sterującego, odbiór odpowiedzi z modułu przekazników (jak i stosowną sygnalizację za pomocą dwubarwnej diody LED) i ponowne uśpienie transceivera oraz mikrokontrolera. Prawda, że proste?



Rysunek 1. Schemat ideowy pilota systemu radioRelay

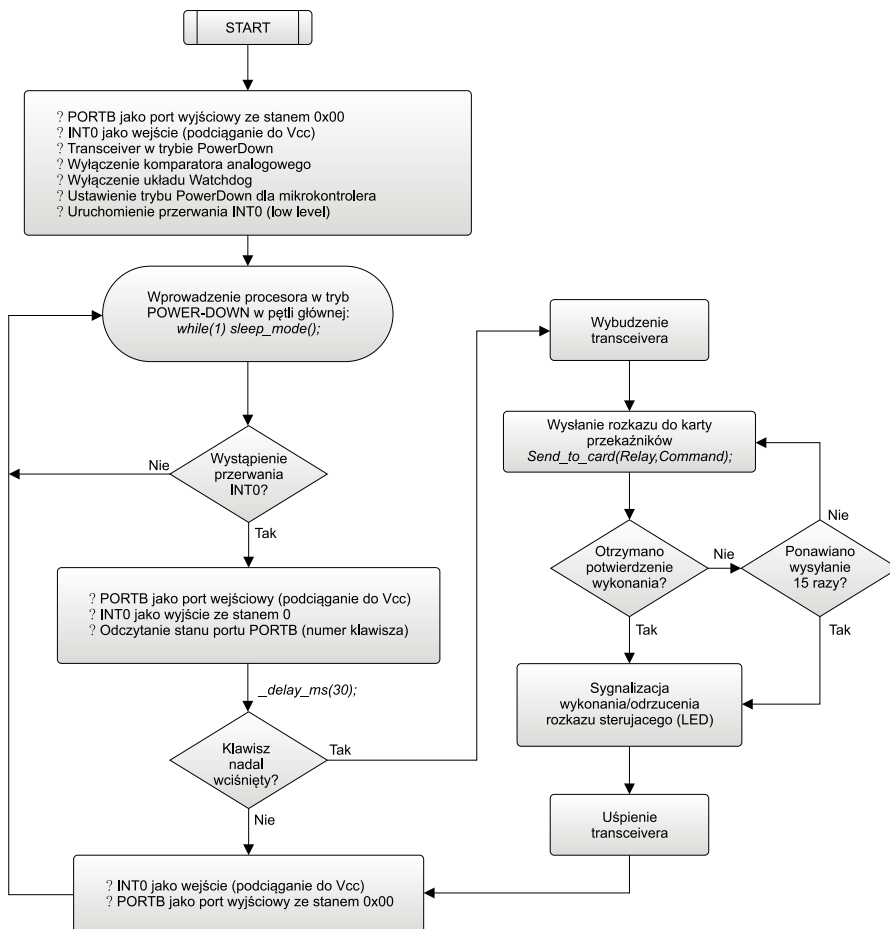
Tylko jak możliwe jest odczytanie stanu klawiszy, skoro PORTB ustawiony był jako wyjściowy (by wciśnięcie jakiegokolwiek przycisku spowodowało wybudzenie procesora dzięki przerwaniu INT0)? To także proste! Tuż po wejściu do procedury obsługi

przerwania INT0, PORTB zostaje skonfigurowany jako port wejściowy z podciąganiem do napięcia zasilania a pin INT0 jako pin wyjściowy z aktywnym stanem 0, dzięki czemu odczyt rejestru PINB daje nam odpowiedź na pytanie o numer wciśniętego klawisza.

Po wykonaniu, odpowiednich czynności i przed wyjściem z procedury obsługi przerwania INT0 zostaje przywrócona domyślna konfiguracja portu PORTB i pinu INT0 w celu umożliwienia dalszej pracy urządzenia. Z resztą najlepszą ilustracją funkcjonalności układu pilota niech będzie kompletny algorytm programu obsługi przedstawiony na **rysunku 2**.

Pozostaje pytanie, jak zabezpieczyć cały system przed potencjalną możliwością realizacji niezamierzonych rozkazów sterujących? To z pozoru trudne zadanie można zrealizować angażując do tego minimum środków, czyli mocy obliczeniowych mikrokontrolera mając na uwadze, iż nie wymagamy tu rozwiązań technologicznych klasy sprzętu wojskowego czy też medycznego. Z pomocą przychodzi nam prosty standard kodowania transmisji danych stosowany powszechnie w pilotach podczerwieni tzw. Manchester oraz odpowiednia konstrukcja samej ramki danych zapewniająca elimina-

REKLAMA



Rysunek 2. Kompletny algorytm programu obsługi pilota układu radioRelay

**Wykaz elementów  
Modułu przekaźników:**

**Rezystory (1/8W):**

R1: 22 kΩ  
R2...R9: 1,5 kΩ

**Kondensatory:**

C1, C3, C5, C6: 100 nF (ceramiczny)  
C2, C4: 100 μF/16 V (elektrolit., niski profil)

**Półprzewodniki:**

U1: 7805  
U2: ATtiny2313 (obudowa DIL20)  
U3: ULN2803 (obudowa DIL18)  
B1: mostek prostowniczy 1A  
CHN1...CHN8: dioda LED 3 mm (czerwona)  
**Inne:**  
TSC – transceiver Aurel RTX-MID-5V  
AC: złącze śrubowe ARK500/2  
OUT1, OUT2: złącze śrubowe ARK500/8  
K1...K8: przekaźnik JZC-49F/12  
TR1: transformator TEZ2.5/D/9V  
L1: dławik 10 μH  
ANTENNA: opis w tekście

**Pilot zdalnego sterowania:**

**Rezystory:** (SMD1206)

R1: 22 kΩ  
R2: 750 Ω

**Kondensatory:** (SMD1206):

C1, C2: 100 nF

**Półprzewodniki:**

U1: ATtiny2313 (SO-20)  
ACK: dioda LED 3 mm, czerwono-zielona ze wspólną katodą  
**Inne:**  
TSC: transceiver Aurel RTX-MID-3V  
S1...S8: przycisk DTSM31  
BATT: metalowy koszyk na baterie CR2450 typu KEYS3009 (Keystone)  
ANTENNA: opis w tekście



cję błędów. Przypomnę krótko podstawowe informacje na temat kodowania typu Manchester. Otóż, kodowanie tego typu zakłada, iż każdy bit transmitowanej ramki danych transmitowany jest w postaci dwóch stanów logicznych, tzw. półbitów, a w środku czasu przeznaczanego na jego przesłanie następuje zmiana tychże stanów. I tak, wysoki poziom logiczny przesyłanego bitu danych kodowany jest poprzez przesłanie półbitu logicznej 1 a następnie logicznego 0, natomiast niski poziom logiczny jest kodowany poprzez przesłanie „półbitu” logicznego 0, a następnie logicznej 1. W przypadku naszego urządzenia przyjmujemy, iż długość trwania półbitu wynosi 200 μs, co daje prędkość transmisji równą 2500 b/s, więc z powodzeniem mieszcząca się w zakresie możliwości układu transceivera. Dodatkowo, przyjmujemy pewne założenia dotyczące konstrukcji ramki transmisji mające na celu eliminację błędnych danych. Pełną ramkę transmisji pokazano na **rysunku 3**.

Każda taka ramka składa się z:

- 4 bitów synchronizacyjnych (0b0000), które są odrzucane przez procedurę dekodującą po stronie odbiornika i służą wyłącznie dopasowania się układu ARW odbiornika do parametrów odbieranego sygnału, zanim nadajnik sygnału wyemituje użyteczną informację,

Synchro	GAP	Header	Command	Relay	~Relay
Bity synchronizacji	Przerwa	Nagłówek	Rozkaz sterujący	Stan przekaźników	Zanegowany stan przekaźników

Rysunek 3. Ramka transmisji nadajnika

```

Listing 1. Treść procedury odpowiedzialnej za wysłanie kompletnej ramki transmisji.
void Send_data(uint32_t Frame_to_send)
{
    // Zaczynamy od bitu Frame_first_bit (MSB) czyli bit nr 23
    register uint8_t Idx;
    //Jeżeli używamy bitów synchronizacyjnych, to wysyłamy 4 takie bity
    #if USE_SYNCHRO_BITS == 1
    for (Idx=0; Idx<4; Idx++)
    {
        RESET_DIN;
        delay_us(Half_bit_period);
        SET_DIN;
        delay_us(Half_bit_period);
    }
    //Podtrzymujemy stan 1 dla restartu ramki i niezgubienia synchronizacji
    delay_us(Half_bit_period*3);
    #endif
    Idx = Frame_first_bit;
    do
    {
        if ( Frame_to_send & ( (uint32_t) 1 << Idx ) )
        {
            SET_DIN; // Dla bitu równego „1” -> najpierw jedynka a potem zero
            delay_us(Half_bit_period);
            RESET_DIN;
            delay_us(Half_bit_period);
        }
        else
        {
            // Dla bitu równego „0” -> najpierw zero a potem jedynka
            RESET_DIN;
            delay_us(Half_bit_period);
            SET_DIN;
            delay_us(Half_bit_period);
        }
    }
    while ( Idx-- != 0 );
    RESET_DIN; //Stan spoczynkowy portu nadawczego
}
    
```

- krótkiej przerwy (3 razy 200 μs) w czasie której jest wysyłana nośna sygnału informująca procedurę dekodującą o nadchodzącym początku ramki transmisji,
- nagłówek (0b1100), który służy przede wszystkim kontroli poprawności emitowanych danych, jak również podtrzymaniu optymalnych ustawień układu ARW odbiornika,
- predefiniowanego rozkazu sterującego typu włącz/wyłącz przekaźnik (4 bity),
- danych związanych z numerem przekaźnika podlegającego zmianie stanu,
- zanegowanych danych jak wyżej (dla kontroli poprawności ramki transmisji).

Treść procedury odpowiedzialnej za wysłanie kompletnej, 24-bitowej ramki transmisji (plus bity synchro) przedstawiono na **listingu 1**.

Jak wspomniano wcześniej i co pokazano na algorytmie obsługi pilota systemu radioRelay, transmisji każdej ramki przeprowadzanej przez układ nadrzędny (pilota) towarzyszy natychmiastowa odpowiedź karty przekaźników, która to zawiera dane oczekiwane przez jednostkę sterującą (potwierdzenie wykonania rozkazu sterującego) jak i służy kontroli poprawności transmisji. Przyjęto, iż przy braku odpowiedzi ze strony karty przekaźników lub w przypadku błędów transmisji, układ nadrzędny, czyli pilot sterujący ponawiał będzie wysyłanie rozkazu sterującego 15 razy. Na **listingu 2** przed-

stawiono treść procedury odpowiedzialnej za utworzenie gotowej do przesłania ramki danych, jej przesłanie, odbiór odpowiedzi i kontrolę poprawności transmisji czyli procedury odpowiedzialnej za komunikację Master<->Slave.

Z powyższych informacji wynika, iż przyjęte rozwiązanie programowe jest dość proste a jednocześnie, co potwierdzono w praktyce, pomimo dość dużej liczby zakłóceń w paśmie 433 MHz, zapewnia doskonałą stopę błędów oraz zmniejsza wymagania sprzętowe i programowe po stronie obu uczestników transmisji. Co prawda można by było zaprzęgnąć do pracy bardziej wyrafinowane algorytmy kontroli poprawności transmisji z mechanizmami CRC8 czy CRC16 włącznie, ale moim zdaniem, byłoby to rozwiązanie przewymiarowane i w żaden sposób nieuzasadnione.

Na koniec opisu urządzenia nadrzędnego (pilota) dodam, iż poprawnemu wykonaniu rozkazu sterującego towarzyszy krótkie mignięcie zielonej diody LED zaś błędowi transmisji towarzyszy odpowiednie krótkie mignięcie czerwonej diody LED. W ten prosty sposób zapewniono realizację założenia projektowego mówiącego o konieczności posiadania informacji zwrotnej o wykonaniu lub odrzuceniu rozkazu sterującego, czyli de facto o stanie przekaźników bezprzewodowej karty wykonawczej. Dodać należy, iż krótkie przyciśnięcie każdego z przycisków

**Listing 2. Treść procedury odpowiedzialnej za komunikację Master↔Slave.**

```

//Gdy wynik działania funkcji != Disconnected, to znaczy,
//że odebrano poprawną ramkę danych
uint8_t Send_to_card(uint8_t Relay, uint8_t Command)
{
    uint32_t SentFrame;
    register uint8_t Trials = 0;
    //Result != Disconnected (0xFF) oznacza, że odebrano poprawną ramkę
    register uint8_t Result = Disconnected;
    //Tworzymy kompletną ramkę danych: Header - Command | Relay | ~ Relay
    // 4 bity 4 bity 8bitów 8bitów
    SentFrame = (uint32_t) (Header | Command) << 16;
    SentFrame |= (uint32_t) Relay << 8;
    SentFrame |= (uint8_t) (~Relay);
    Pwrdwn_to_tx(); //Przechodzimy do nadawania
    while (!(Result == Disconnected) && (Trials < 15))
    {
        Send_data(SentFrame); //Wysyłamy przygotowaną ramkę
        TCCR1B &= ~(1<<ICES1); // Zbocze opadające na ICP
        //Uruchomienie przerwania ICP (w pliku Aurel.c nie
        //jest uruchamiane domyślnie)
        TIMSK |= (1<<ICIE1);
        //Przełączamy transceiver w tryb odbioru - 300us
        Tx_to_rx();
        //Odczekujemy pewien czas by karta przełączników
        //odpowiedziała nam na wysłany rozkaz
        _delay_ms(13);
        //Dezaktywujemy przerwanie ICP
        TIMSK &= ~(1<<ICIE1);
        //Po tym czasie transceiver przełączamy z powrotem
        //do nadawania bo jest to jego stan domyślny - 405us
        Rx_to_tx();
        //Karta przełączników odpowiedziała,
        //a odpowiedź umieszczona jest w zmiennej Frame
        if (Frame_ready)
        {
            //Slave odpowie: 1 -> jeżeli wykonał komendę lub
            //Disconnected -> jeśli zdekoduje błędną komendę
            if (Frame == 1) Result = Frame;
            Frame_ready = 0;
        }
        Trials++;
        _delay_us(400);
    }
    RESET_ENABLE; //Przechodzimy do trybu PoweDown
    //Zerujemy chociaż nie przechodzimy do RX ale by mieć stan wyjściowy portów
    sterujących
    RESET_TX_RX;
    return Result;
}

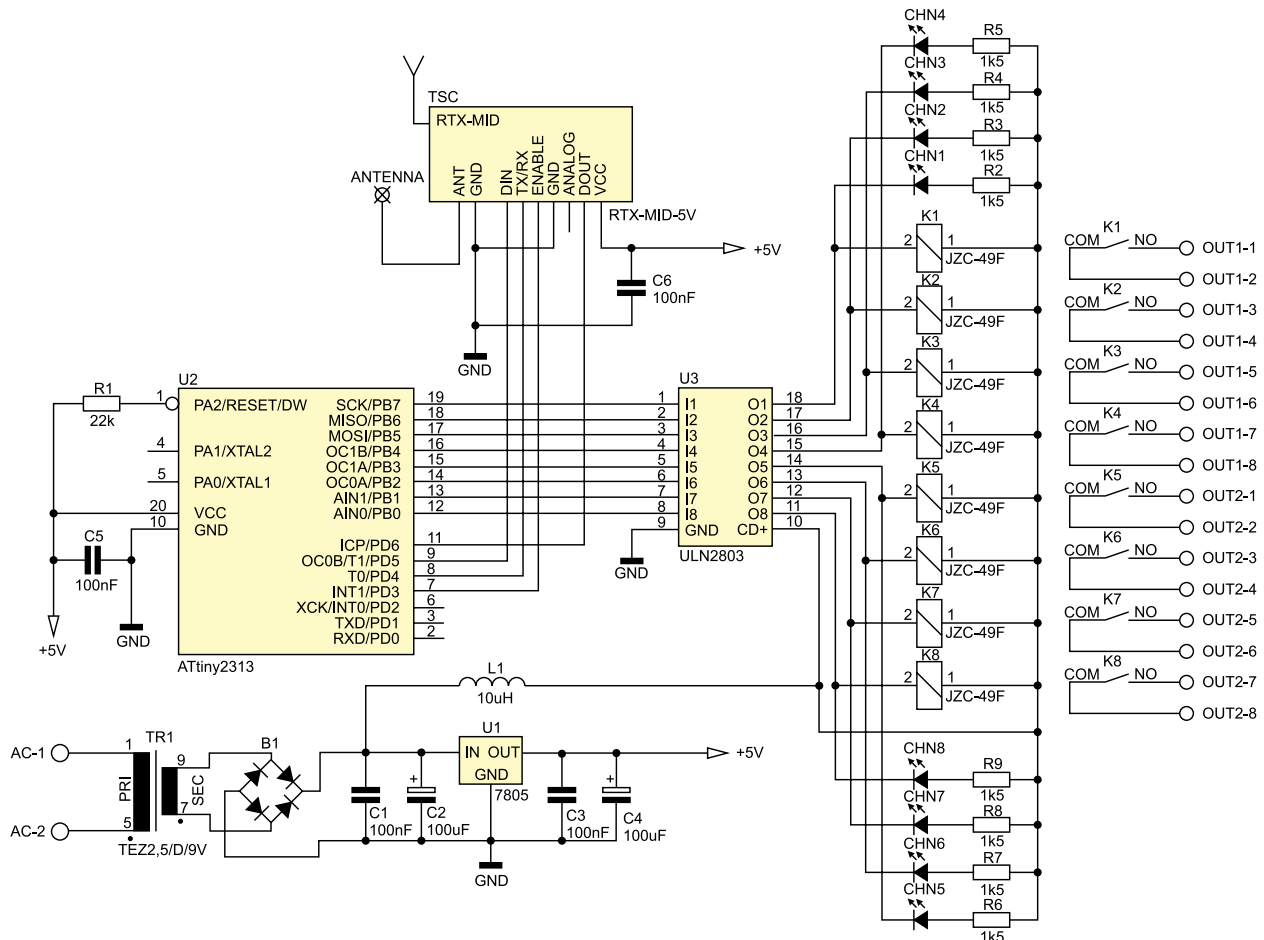
```

powoduje wysłanie rozkazu na wyłączenie odpowiedniego przełącznika zaś długie przyciśnięcie (ok. 500 ms) powoduje wysłanie rozkazu załączającego odpowiedni przełącznik.

### Odbiornik – moduł z przełącznikami

Przejdźmy, zatem do schematu urządzenia wykonawczego, jakim jest bezprzewodowy moduł przełączników pokazany na **rysunku 4**. Tym razem do czynienia mamy z nieco bardziej rozbudowanym systemem mikroprocesorowym, choć stopień jego komplikacji wynika wyłącznie z niezbędnej liczby elementów potrzebnej do sterowania 8 przełącznikami mocy.

Jak poprzednio, sercem urządzenia jest mikrokontroler ATtiny2313 odpowiedzialny wyłącznie za obsługę wbudowanego transceivera, domyślnie pracującego w trybie odbiornika oraz sterowanie (za pomocą układu ULN2803) pracą wspomnianego zespołu 8 przełączników wykonawczych. Zasada działania tego układu jest bardzo prosta: moduł ten pracujący domyślnie w trybie odbiornika oczekuje na ważną dla niego ramkę danych sterujących (zawierającą odpowiedni rozkaz jak i cechującą się odpowiednią budową) przesyłaną przez moduł sterujący (*pilot*). Po otrzymaniu takiej ramki danych, program obsługi urządzenia przełącza transceiver



Rysunek 4. Schemat ideowy bezprzewodowej karty przełączników systemu radioRelay

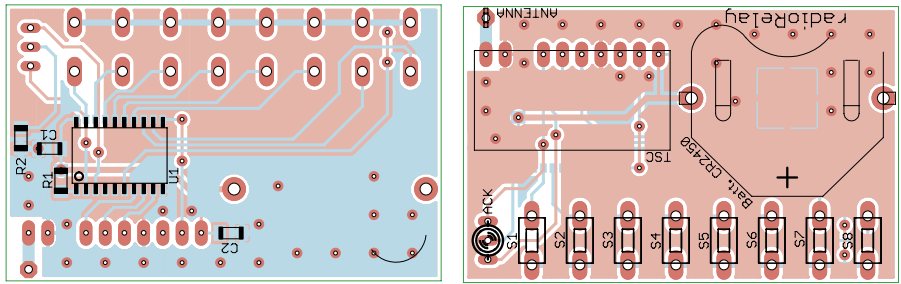
**Ustawienia ważniejszych FUSE BIT'ów mikrokontrolera**

Moduł przekaźników :  
 CKSEL3..0: 0100  
 SUT1..0: 10  
 CKDIV8: 1  
 Pilot zdalnego sterowania  
 CKSEL3..0: 0100  
 SUT1..0: 10  
 CKDIV8: 1  
 DWEN: 1  
 WDTON: 1

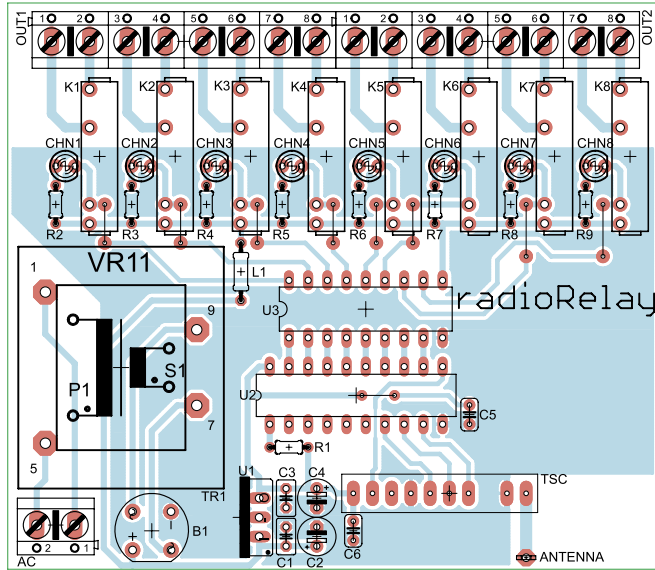
w tryb nadajnika, następnie odsyła natychmiast potwierdzenie wykonania rozkazu sterującego służące jednocześnie kontroli poprawności transmisji, po czym przechodzi ponownie w tryb odbiornika. Tym razem, nie korzystamy z trybów obniżonego poboru mocy zarówno po stronie modułu transceivera jak i po stronie mikrokontrolera zwłaszcza, iż w projekcie urządzenia przewidziano zbudowanie kompletnego układu zasilanego napięciem sieciowym 230 V AC. To zadanie byłoby trudne do realizacji, gdyż jak już wspomniano wcześniej, transceiver karty przekaźników pracuje domyślnie w trybie odbiornika nasłuchując czy nie są przesyłane dane. Na koniec, wspomnę tylko, iż równolegle do cewek przekaźników mocy dołączono diody LED sygnalizujące stan pracy wspomnianych elementów mocy.

**Montaż**

Schemat montażowy pilota zdalnego sterowania systemu radioRelay pokazano na na **rysunku 5**. Jego montaż rozpoczynamy od przylutowania mikrokontrolera. Do jego zamontowania wystarczy zwykła stacja lutownicza. Następnie lutujemy pozostałe elementy SMD (mając na uwadze odpowiednią stronę montażu), a na końcu przyciski, moduł transceivera, gniazdo baterii CR2450 oraz dwubarwną diodę LED. Jako anteny można użyć zwykłego kawałka drutu miedzianego o przekroju 0,5 mm<sup>2</sup> i długości około 17,2 cm. Co ciekawe, testy praktyczne pokazały, że czasami zastosowanie zwykłego kawałka drutu daje lepszy efekt, niż użycie anten SMD (oczywiście zależy to w dużym stopniu od projektu samej płytki drukowanej). Długość takiej anteny można zmniejszyć do około 8,3 cm poprzez uformowanie cewki powietrznej położonej u podstawy anteny. Należy jednak pamiętać, iż anteny nadajnika i odbiornika powinny być tego samego rodzaju oraz znajdować się w położeniu prostopadłym do powierzchni



**Rysunek 5. Schemat montażowy pilota systemu radioRelay**



**Rysunek 6. Schemat montażowy bezprzewodowej karty przekaźników systemu radioRelay**

masy poszczególnych płytek drukowanych (w przypadku anten wykonanych z kawałka drutu). Ich wzajemne położenie wpływa na zdecydowanie na zasięg kompletu urządzeń.

Na **rysunku 6** pokazano schemat montażowy modułu wykonawczego, tj. modułu z przekaźnikami. Tym razem do czynienia mamy z jednowarstwową płytką drukowaną o dość prostej strukturze przeznaczoną wyłącznie do montażu przewlekanego. Montaż ten rozpoczynamy od wlotowania wszystkich zworek (uwaga: jedna z nich znajduje się pod mikrokontrolerem). Następnie, lutujemy elementy bierne, złącza, półprzewodniki (mikrokontroler najlepiej włożyć w podstawkę), a na końcu transformator zasilający, przekaźniki oraz moduł transceivera. Należy także zwrócić szczególną uwagę na konieczność pocynowania grubą warstwą cyny ścieżek przewodzących duże prądy tj. ścieżek pomiędzy złączami śrubowymi OUT1...

OUT2 a przekaźnikami K1...K8 (dla tych ścieżek nie stosowano solder-maski). Antenę wykonujemy według tych samych zasad jak dla modułu pilota sterującego. Poprawnie zmontowany układ powinien działać tuż po podłączeniu zasilania.

**Uwaga! Na płycie karty przekaźników zamontowano kompletny zasilacz sieciowy, łącznie z transformatorem zasilanym napięciem sieciowym 230 V AC oraz zamontowano elementy będące na potencjale tegoż napięcia. Istnieje niebezpieczeństwo porażenia elektrycznego, które stanowi zagrożenie dla życia i zdrowia użytkowników. W związku z tym, montaż układu w tym zakresie powierzyć należy osobie posiadającej uprawnienia elektryczne w zakresie eksploatacji urządzeń o napięciu do 1 kV oraz niezbędną wiedzę i doświadczenie.**

**Robert Wołgajew, EP**

REKLAMA

**Przekaźnikowy wyłącznik czasowy  
 AVT1689**

[www.sklep.avt.pl](http://www.sklep.avt.pl)

