radioDimmer

Regulator oświetlenia w mieszkaniu

AVT 5376

Prezentujemy projekt radioDimmera, który umożliwia scentralizowaną, bezprzewodową, wielopunktową regulację natężenia oświetlenia. Jest to system, będący siecią adresowalnych, bezprzewodowych modułów wykonawczych (slave) realizujących funkcję ściemniacza, zarządzanych z jednej jednostki głównej (master) wyposażonej w rozbudowany interfejs użytkownika.

Rekomendacje: *rozbudowany system regulowania natężenia światła, który przyda się w mieszaniu lub przy aranżowaniu wnętrza, komponowaniu sceny itp.*

W planach każdego elektronika są zapewne takie projekty, które z tych czy innych względów powędrowały na przysłowiową półkę w oczekiwaniu na tzw. „lepsze czasy". Sam, z nieskrywanym sentymentem, przypominam sobie czasy „Młodego technika", w którym to po raz pierwszy spotkałem się z projektami urządzeń pracujących z wykorzystaniem transmisji radiowej i które to wtedy były dla mnie (i nie tylko) z wielu względów nieosiągalne. Dzisiaj, w dobie ogólnodostępnej wiedzy oraz ciągłego postępu w dziedzinie gotowych rozwiązań związanych z bezprzewodową transmisją danych, realizacja wielu wspomnianych projektów staje się trywialna z punktu widzenia sprzętu i niezbędnych nakładów finansowych. Aplikacją, której krótka historia sięga wspomnianych czasów jest projekt radioDimmera, który w swoich założeniach miał udostępniać możliwość scentralizowanej, bezprzewodowej i wielopunktowej regulacji natężenia oświetlenia. Dzisiaj, dzięki zastosowaniu gotowych modułów transceiverów pracujących w paśmie ISM oraz mikrokontrolerów, budowa takiego systemu okazuje się bardzo łatwa. Opisywany projekt regulatora natężenia oświetlenia jest siecią adresowalnych, bezprzewodowych modułów wykonawczych realizujących funkcję ściemniacza, zarządzanych z jednej jednostki głównej (*Master*) wyposażonej w rozbudowany interfejs użytkownika.

Zacznijmy od opisu modułu transmisyjnego, którym jest w tym wypadku gotowy moduł transceivera typu RTX-MID-5V produkowany przez firmę Aurel. Chętnie stosuję go w swoich projektach, ponieważ jest łatwy w użyciu, a przy tym niezawodny. Przypomnijmy pokrótce najważniejsze parametry użytkowe tego ciekawego modułu, ponieważ zapewne nie każdy z Czytelników stosował go w swoich projektach:

- praca w trybie half-duplex,
- modulacja ASK,
- bardzo krótkie czasy przełączania pomiędzy trybami nadajnika, odbiornika oraz PowerDown (przy zachowaniu zależności czasowych przedstawionych w dokumentacji producenta),
- moc wyjściowa 10 mW,
- bardzo niski pobór mocy w trybie Power Down (rzędu 1 μA),
- maksymalna prędkość transmisji 9600 bps,
- bardzo wysoka czułość toru odbiornika (–106 dBm),
- niewielkie wymiary zewnętrzne.

Wygląd modułu transceivera oraz rozmieszczenie jego wyprowadzeń zamieszczono w ramce.

Użyty moduł transmisyjny nie wymaga konfigurowania, jak to ma miejsce w wypadku innych produktów, na przykład transceiverów RFM12 firmy HOPE RF, o których na wszelkiego rodzaju forach internetowych poświęconych elektronice krążą niemal legendy. Nie bez znaczenia w naszym przypadku były także niewielkie wymiary zewnętrzne modułów firmy Aurel. Co oczywiste, w jego miejsce można byłoby zastosować dowolny element tego typu spełniający niewygórowane wymagania opisywanego systemu.

Przejdźmy, zatem do szczegółów implementacyjnych dotyczących wspomnianej transmisji, jako że jest to zagadnienie niezmiernie ciekawe. Wspomniałem już, iż im-

W ofercie AVT*
AVT-5376/1 MASTER A AVT-5376/1 SLAVE A
AVT-5376/1 MASTER UK AVT-5376/1 SLAVE UK

Podstawowe informacje:

Master:
- Napięcie zasilania: 9…12 V DC.
- Prąd zasilający (tryb bezczynności/praca): 20/80 mA.
- Liczba obsługiwanych modułów Slave: 16.
- Długość nazwy modułu Slave: 8 znaków.
- Zakres regulacyjny jasności: 0…100% z krokiem co 2%.
- Częstotliwość pracy transceivera: 433,92 MHz.
- Zasięg w terenie otwartym: ok. 100 m

Slave:
- Napięcie zasilania: 230 V AC.
- Moc pobierana (tryb bezczynności/praca): poniżej 1 W.
- Liczba możliwych adresów sprzętowych układu Slave: 16.
- Zakres regulacyjny jasności: 0…100% z krokiem co 2%.
- Częstotliwość pracy transceivera: 433.92 MHz.
- Zasięg w terenie otwartym: ok. 100 m

Ustawienia ważniejszych FUSE BIT'ów
- radioDimmer_Master: CKSEL3..0: 0100, SUT1..0: 10, EESAVE: 0
- radioDimmer_Slave: CKSEL3..0: 0100, SUT1..0: 10, EESAVE: 0, CKDIV8: 1

Dodatkowe materiały na CD/FTP:

ftp://ep.com.pl, user: 13621, pass: 175brjf7
- wzory płytek PCB
- karty katalogowe i noty aplikacyjne elementów oznaczonych w Wykazie elementów kolorem czerwonym

Projekty pokrewne na CD/FTP:

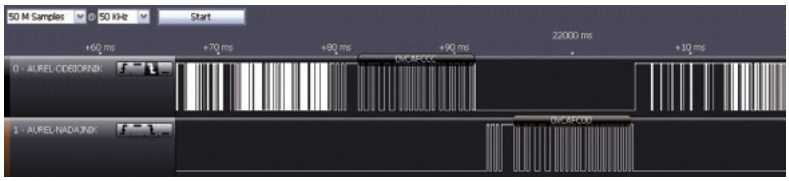
(wymienione artykuły są w całości dostępne na CD)

AVT-5361 4dimmer - 4-kanałowy regulator oświetlenia (EP 9/2012)
AVT-5336 Sterownik oświetlenia sufitu (EP 3/2012)

* 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 wymienionych w załączniku pdf
AVT xxxx C	to nic innego jak zmontowany zestaw B, czyli elementy wlutowane 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**



Rysunek 1. Rzeczywiste przebiegi sygnałów sterujących po stronie układu Slave dla przypadku zapytania o bieżące ustawienia jasności (rozkaz 0b1100)

Rysunek 2. Rzeczywiste przebiegi sygnałów sterujących po stronie układu Slave dla rozkazu ustawienia nowego poziomu jasności (rozkaz 0b0011).

Listing 1 Treść procedury odpowiedzialnej za wysłanie kompletnej ramki transmisji

```
void Send_data(uint32_t Frame_to_send)
{
  register uint8_t Idx;
  //Jeżeli używamy bitów synchronizacyjnych,
  //to wysyłamy 4 bity o wartości 0
  #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 nie zgubienia synchronizacji
  _delay_us(Half_bit_period*3);
  #endif

  //Zaczynamy od bitu Frame_first_bit (MSB), bit nr 23
  Idx = Frame_first_bit;
  do
  {
    if ( Frame_to_send & ( (uint32_t) 1 << Idx ) )
    {
      SET_DIN;
//Dla bitu równego „1" -> najpierw 1, a potem 0
      _delay_us(Half_bit_period);
      RESET_DIN;
      _delay_us(Half_bit_period);
    }
    else
    {
      RESET_DIN;  // Dla bitu równego 0 - najpierw 0 a potem 1
      _delay_us(Half_bit_period);
      SET_DIN;
      _delay_us(Half_bit_period);
    }
  }
  while ( Idx-- != 0 );
  RESET_DIN;  //Stan spoczynkowy portu nadawczego
}
```

plementacja modułu tego typu w docelowym systemie mikroprocesorowym jest niezmiernie prosta i może nawet sprowadzać się do podłączenia odpowiednich wyprowadzeń In/Out modułu do interfejsu USART mikrokontrolera (przynajmniej według zapewnień producenta). Rozwiązanie takie nie jest jednak pozbawione wielu wad, a co najgorsze – nie gwarantuje małej wartości stopy błędów. Wynika to z kilku zjawisk charakterystycznych dla tego typu rozwiązań, o których nie każdy pamięta.

Po pierwsze, odbiornik modułu transceivera potrzebuje trochę czasu, aby dostroić wewnętrzne układy ARW do parametrów odbieranego sygnału. Powoduje to, iż początkowa część odbieranej ramki transmisji może zostać utracona i/lub zniekształcona. Po drugie, odbiornik nie został zabezpieczony przed ogólnie występującymi zburzeniami transmisji radiowej, zatem po stronie systemu mikroprocesorowego stoi zadanie odróżniania „śmieci" od ważnych ramek danych. Jedyna pomoc ze strony modułu, na którą można liczyć, to realizacja funkcji *squelch* (doprowadzenie 8 transceivera), która powoduje ściąganie do masy wyjścia odbiornika w przypadku braku nośnej sygnału, lecz jednocześnie niesie za sobą konsekwencje w postaci zmniejszenia czułości odbiornika o 3 dB. Ponadto, ten sposób nie rozwiązuje wszystkich problemów związanych z potencjalnym wpływem zaburzeń na transmitowane dane, a co za tym idzie – nie zapewnia małej stopy błędów. Jak w takim razie zabezpieczyć się przed błędnymi danymi? Odpowiedź wydaje się bardzo prosta, a dodatkowo poparta praktyką wielu firm produkujących scalone kodery/dekodery do zastosowań bezprzewodowych. Najlepiej zastosować jakiś sprawdzony i odporny na błędy system kodowania danych, który zminimalizuje ryzyko błędów transmisji i jednocześnie sprawi, iż nasz system stanie się odporny na zakłócenia występujące w użytecznym paśmie. Jednym z takich systemów bardzo prostych w implementacji, a jak pokazuje codzienna praktyka, zarazem odpornym na zakłócenia, jest system kodowania danych typu Manchester (kodowanie bifazowe sygnału cyfrowego) szeroko stosowany w układach sterowania z wykorzystaniem podczerwieni (np. standard RC5). Kodowanie tego typu zakłada, iż każdy bit transmitowanej ramki danych kodowany jest w postaci dwóch stanów logicznych, tzw. półbitów, a w środku czasu przeznaczonego na jego przesłanie następuje zmiana tychże stanów. I tak, wysoki stan logiczny przesyłanego bitu danych kodowany jest poprzez przesłanie „półbitu" logicznej jedynki a następnie logicznego zera natomiast niski stan logiczny kodowany jest poprzez przesłanie „półbitu" logicznego zera a następnie logicznej jedynki (tak jest w przypadku naszego układu). W przypadku naszego urządzenia przyjmiemy, iż długość trwania półbitu wynosi 200 µs, co daje prędkość transmisji równą 2500 bps a więc z powodzeniem mieszcząca się w zakresie możliwości układu transceivera. Dodatkowo przyjmujemy pewne założenia dotyczące

konstrukcji ramki transmisji mające na celu eliminację błędnych danych. Ramka transmisji powinna mieć następującą postać:

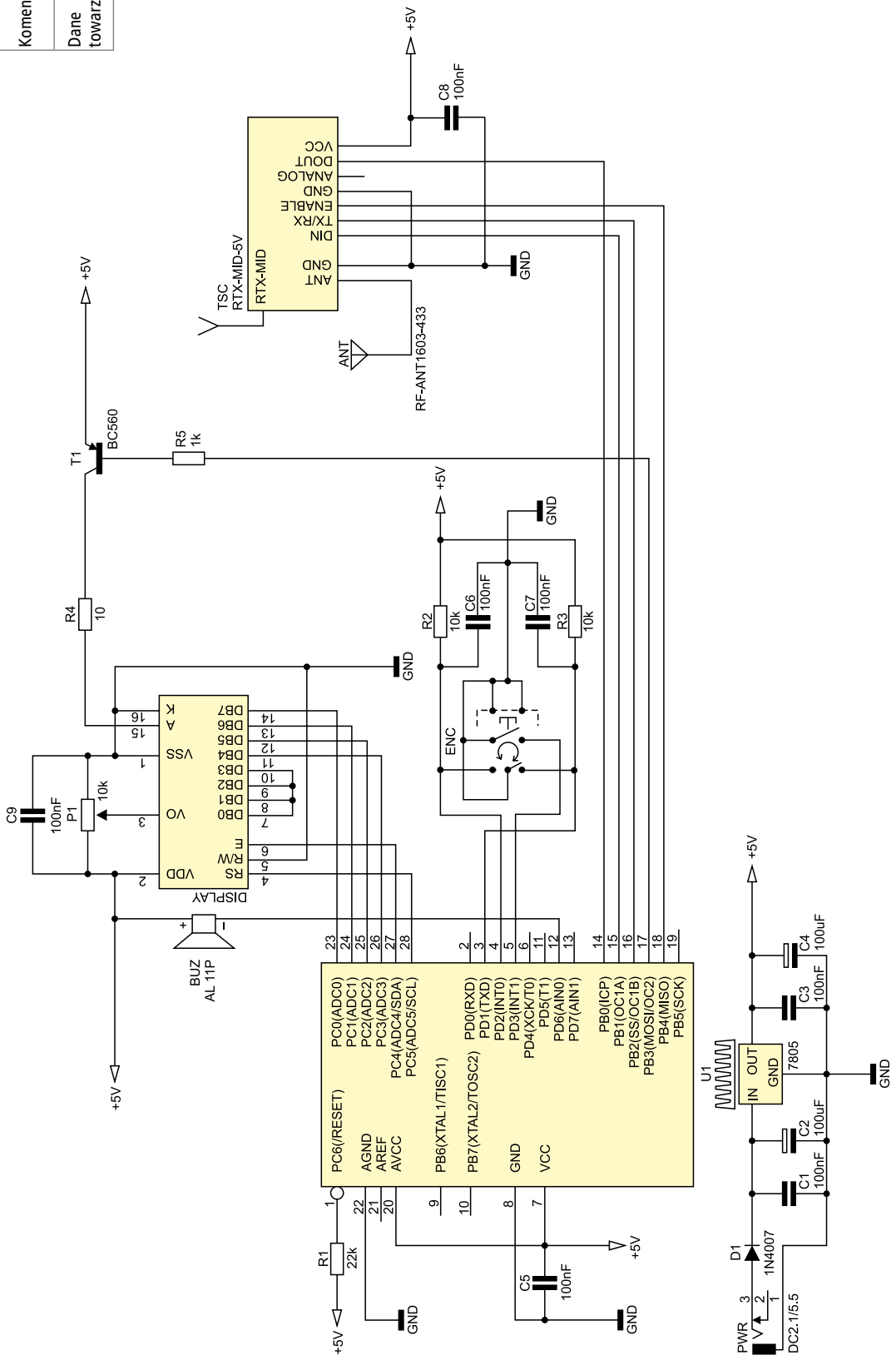
Synchro	GAP	Header	Address	Command	Data
Bity synchro	Przerwa	Nagłówek	Adres	Komenda	Dane towarzyszące

Każda taka ramka składa się z:

- 4 bitów synchronizacyjnych (0b0000), które odrzucane są przez procedurę dekodującą po stronie odbiornika i służą wyłącznie celom dopasowania się układu ARW odbiornika do parametrów odbieranego sygnału zanim nadajnik wyemituje użyteczną informację,
- krótkiej przerwy (3×200 µs), w czasie której jest emitowana nośna sygnału, a która informuje procedurę dekodującą o nadchodzącym początku ramki transmisji,
- nagłówka (0b11001010), który służy przede wszystkim kontroli poprawności emitowanych danych jak również podtrzymaniu optymalnych ustawień układu ARW odbiornika,
- adresu modułu wykonawczego (4 bity),
- rozkazu sterującego (4 bity),
- danych związanych bezpośrednio z rodzajem rozkazu sterującego.

Treść procedury odpowiedzialnej za wysłanie kompletnej, 24-bitowej ramki transmisji (plus bity synchronizacyjne) zgodnej z powyższymi założeniami przedstawiono na **listingu 1**.

Co ważne, transmisji każdej ramki przeprowadzanej przez układ Master towarzyszy natychmiastowa odpowiedź układu *Slave*, która to zawiera dane oczekiwane przez jednostkę zarządzającą systemem (*Master*) jak i służy kontroli poprawności transmisji. W ten prosty sposób układ nadrzędny kontroluje także obecność znalezionych wcześniej i zapamiętanych układów *Slave*. W przypadku braku odpowiedzi ze strony układu *Slave*, układ *Master* ponawia stosowne zapytanie (do 15 razy), po czym, w przypadku dalszego braku odpowiedzi, blokuje możliwość sterowania układem podrzędnym (przy użyciu interfejsu użytkownika układu *Master*, do czasu najbliższego nawiązania łączności) sygnalizując tym samym problem transmisji z wybranym modułem. Kontrola obecności układów *Slave*, wyłącznie tych, które przestały odpowiadać na wysyłane rozkazy sterujące, wykonywana jest w pętli głównej programu obsługi



Rysunek 3. Schemat ideowy modułu Master systemu radioDimmer.

Tabela 1. Lista dostępnych rozkazów sterujących systemu radioDimmer (Dummy_byte=0xCC)

Rozkaz	Opis	Wartość bajta danych układu Master	Odpowiedź układu Slave	Znaczenie bajta danych odpowiedzi układu Slave
0b0011 (Set_phase)	Ustawia jasność świecenia	Jasność świecenia (≤50)	Ramka danych bez zmian lub zmieniony ostatni bajt w przypadku błędu.	Wysłana wartość jasności świecenia lub Dummy_byte w przypadku błędu transmisji.
0b1100 (Ask_for_phase)	Zapytanie o wartość jasności świecenia	Dummy_byte	Odsyła zmodyfikowaną ramkę transmisji. Zmianie ulega wartość bajta danych.	Aktualna wartość jasności świecenia lub Dummy_byte w przypadku błędu transmisji.

sterownika *Master* w cyklu 1-sekundowym. Aby dopełnić opisu oprogramowania systemu radioDimmer, na **listingu 2** przedstawiono treść procedury odpowiedzialnej za utworzenie gotowej do przesłania ramki danych, jej przesłanie, odbiór odpowiedzi i kontrolę poprawności transmisji czyli procedury odpowiedzialnej za komunikację pomiędzy modułami *Master* i *Slave*.

W **tabeli 1** umieszczono listę dostępnych rozkazów sterujących wraz z opisem ich znaczenia jak również specyfikację oczekiwanej odpowiedzi ze strony układu *Slave* (także z opisem jej znaczenia). Należy podkreślić, iż bity definiujące konkretny rozkaz sterujący dobrano w taki sposób, iż przekłamanie na jednej pozycji nie spowoduje błędnej interpretacji numeru rozkazu, co dodatkowo zabezpiecza urządzenie przed wykonaniem niezamierzonych ustawień.

Jako uzupełnienie informacji podanych powyżej, na **rysunku 1** i **rysunku 2** pokazano rzeczywiste oscylogramy sygnałów sterujących po stronie układu *Slave* (odbiornika i nadajnika wbudowanego transceivera) dla przypadku zapytania wysłanego przez układ *Master* o bieżące ustawienia jasności jak i rozkazu ustawienia nowego poziomu jasności (adres badanego układu 0x0F). Przy okazji, na rysunkach tych widać sporą liczbę zakłóceń (po stronie odbiornika) przed i po odebraniu użytecznej ramki danych, które to z powodzeniem odrzucane są przez procedurę dekodującą.

Jak widać, przyjęte rozwiązanie jest dość proste jednakże, co potwierdzono w praktyce, pomimo dość dużej liczby zakłóceń w paśmie 433 MHz, zapewnia ono doskonałą stopę błędów a jednocześnie zmniejsza wymagania sprzętowe jak i programowe po stronie układu *Slave*. Można byłoby co prawda zaprzęgnąć do pracy bardziej wyrafinowane algorytmy kontroli poprawności transmisji z mechanizmami CRC8 czy CRC16 włącznie, ale moim zdaniem, byłoby to przysłowiowe „strzelanie z działa do muchy" wszak nie sterujemy sprzętem wojskowym czy medycznym tylko budujemy prosty sterownik oświetlenia. Oczywiście, by układ *Master* (sterownik zarządzający) w ogóle mógł współpracować z jakimkolwiek układem *Slave* (modułem wykonawczym) stanowiącym wspomnianą bezprzewodową sieć sterującą, musi znać adresy wszystkich, aktywnych modułów tego typu jak i ich bieżące nastawy. Do tego celu przewidziano zautomatyzowaną procedurę konfiguracyjną wywoływaną z poziomu interfejsu użytkownika układu *Master*, którą opiszę w dalszej części artykułu.

Przejdźmy, zatem do schematu układu *Master*, który to zamieszczono na **rysunku 3**. Jak widać, jest to dość prosty system mikroprocesorowy, którego „sercem" a zarazem elementem odpowiedzialnym za realizację założonej funkcjonalności jest mikrokontroler ATmega8. Co więcej, w prezentowanym urządzeniu zastosowano interesujący interfejs użytkownika czyniąc, mam nadzieję, jego obsługę prostą i intuicyjną. Otóż, za regulację wszystkich parametrów systemu odpowiedzialny jest wyłącznie jeden element – enkoder ze zintegrowanym weń przyciskiem. Uważny czytelnik zauważy zapewne, iż jedno z wyjść enkodera podłączone jest do portu mikrokontrolera oznaczonego INT0 wywołującego obsługę jednego z przerwań zewnętrznych. Właśnie wspomniana procedura realizuje obsługę tego wygodnego elementu regulacyjnego dekodując kierunek obrotów a następnie, posiłkując się wartością flagi ustawianej w pętli głównej aplikacji, zmienia wartość wybranego parametru. Drugim, ciekawym elementem interfejsu użytkownika jest sam wyświetlacz. Jest to moduł o organizacji 4×20 znaków wykonany w technologii OLED charakteryzujący się doskonałymi parametrami optycznymi. Przyznam szczerze, iż coraz częściej w swoich projektach stosuję alfanumeryczne „odpowiedniki" typowych wyświetlaczy ale wykonane w technologii OLED. Ich parametry optyczne są imponujące zaś cena w tej chwili nie odbiega drastycznie od cen elementów tradycyjnych tzn. wykonanych w technologii LCD z podświetleniem LED. Dla „tradycjonalistów", przewidziano możliwość zastosowania typowego wyświetlacza LCD z podświetleniem LED a dzięki zastosowaniu wbudowanego w strukturę mikrokontrolera układu czasowo-licznikowemu Timer2 pracującemu w trybie generatora PWM (końcówka OC2/PB3), możliwa stała się realizacja funkcji automatycznego przyciemniania jego

Uwaga! Na płytce urządzenia Slave zamontowano kompletny zasilacz łącznie z transformatorem zasilanym napięciem sieciowym 230 V AC oraz elementy będące na potencjale tegoż napięcia. Istnieje niebezpieczeństwo porażenia prądem elektrycznym o napięciu 230 V AC, co może stanowić 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.

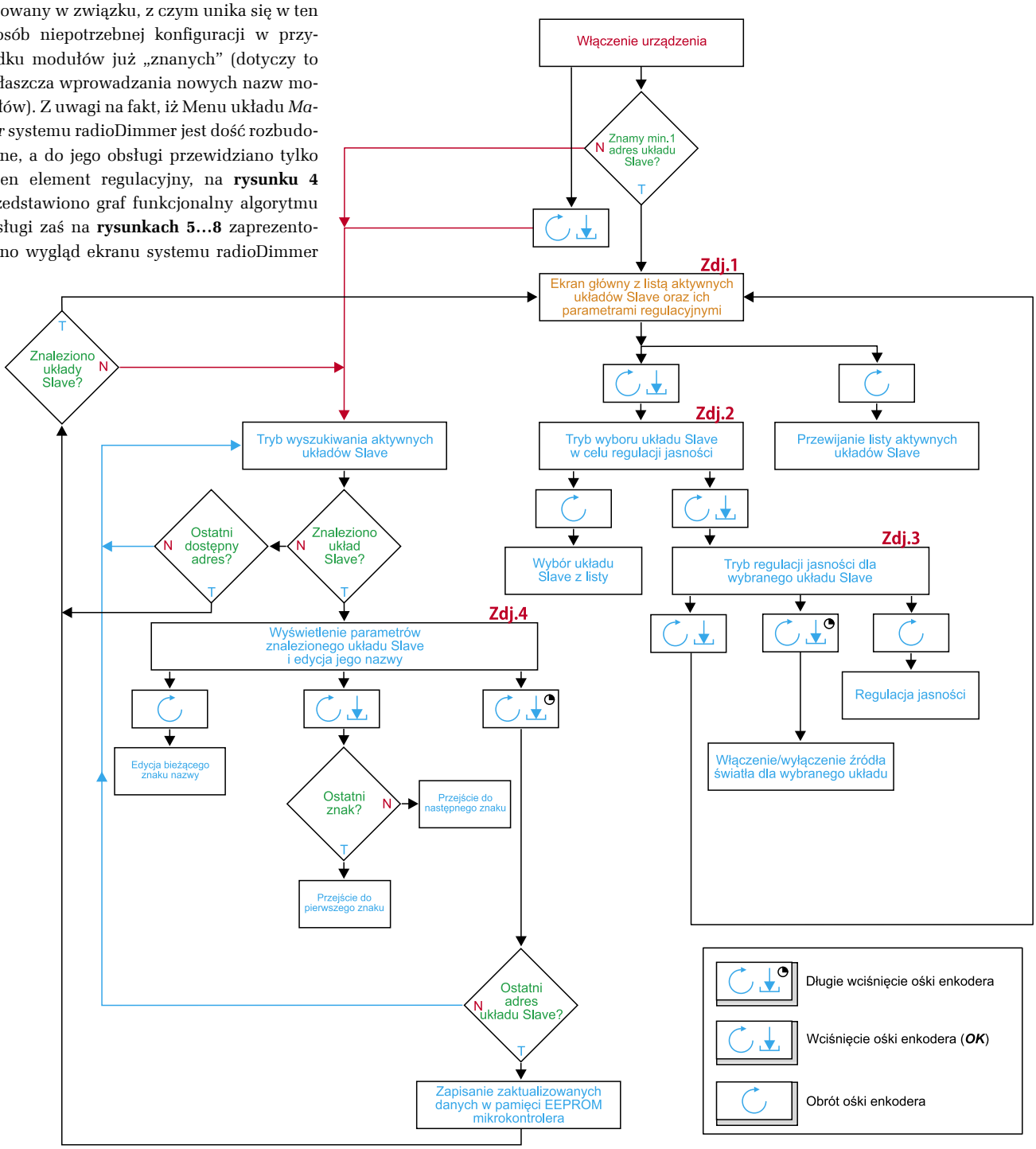
Wykaz elementów

Moduł Master
Rezystory:
R1: 22 kΩ
R2, R3: 10 kΩ
R4: 10 Ω (lub zworka)
R5: 1 kΩ
P1: potencjometr montażowy 10 kΩ
Kondensatory:
C1, C3, C5…C9: ceramiczny 100 nF
C2, C4: elektrolityczny 100 µF/16V (niski profil)
Półprzewodniki:
U1: 7805
U2: ATmega8 (obudowa DIP28)
T1: BC560
D1: 1N4007
Inne:
DISPLAY: wyświetlacz OLED 4×20 znaków typu WEH002004ALPP5N00000 (sterownik zgodny z HD44780)
TSC: transceiver Aurel RTX-MID-5V
PWR: gniazdo zasilające DC2.1/5.5
BUZ: buzzer piezzoelektryczny 5V
ENC: enkoder ze zintegrowanym przyciskiem
ANT: antena SMD RF-ANT1603-433 lub odcinek drutu miedzianego (patrz opis montażu)

Moduł Slave
Rezystory: (miniaturowe 1/8 W)
R1, R2: 68 kΩ/2 W
R3: 22 kΩ
R4: 510 Ω
R5: 33 kΩ
R6: 390 Ω
R7: 360 Ω
R8: 39 Ω
Kondensatory:
C1, C3, C5, C6: 100 nF (ceramiczny)
C2, C4: 100 µF/16 V (elektrolityczny, niski profil)
C7: 10 nF/400 V
Półprzewodniki:
U1: 78L05
U2: ATtiny2313 (obudowa SO20)
OK1: PC814 (DIP4)
OK2: MOC3023 (DIP6)
T1: BT138 (TO220)
B1: B05S (mostek prostowniczy SMD)
TSC: transceiver Aurel RTX-MID-5V
RX: dioda LED 3 mm, czerwona
Inne:
LOAD, AC – złącze śrubowe typu AK500/2
ADDR: nastawnik kodowy MCRH4AF-16R
TR1: transformator SMD typu TEZ-0.5/D/9V
ANT: RF-ANT1603-433433, antena SMD lub odcinek drutu miedzianego (opis w tekście)

podświetlenia. Intensywność tego podświetlenia podlega ograniczeniu po czasie około 30 sekund bezczynności urządzenia (braku jakichkolwiek działań ze strony użytkownika). Jak wspomniano wcześniej, przycisk wbudowany w oskę enkodera realizuje szereg funkcji kontrolno-regulacyjnych. Jedną z nich jest uruchomienie specjalnej procedury konfiguracyjnej, której zadaniem jest wyszukanie wszystkich aktywnych układów *Slave*, nadanie im własnych nazw (korzystając z interfejsu użytkownika) w celu dalszej, łatwej identyfikacji oraz określenie parametrów regulacyjnych. Procedura ta sprawdza także czy wyszukany moduł *Slave* (dokładnie jego adres sprzętowy) był już wcześniej logowany w związku, z czym unika się w ten sposób niepotrzebnej konfiguracji w przypadku modułów już „znanych" (dotyczy to zwłaszcza wprowadzania nowych nazw modułów). Z uwagi na fakt, iż Menu układu *Master* systemu radioDimmer jest dość rozbudowane, a do jego obsługi przewidziano tylko jeden element regulacyjny, na **rysunku 4** przedstawiono graf funkcjonalny algorytmu obsługi zaś na **rysunkach 5...8** zaprezentowano wygląd ekranu systemu radioDimmer dla wybranych trybów pracy układu. W tym miejscu przejdźmy do omówienia układu *Slave* systemu radioDimmer, którego schemat przedstawiono na **rysunku 9**.

Tym razem do czynienia mamy z jeszcze prostszym systemem mikroprocesorowym, który do realizacji postawionych mu zadań wykorzystuje prosty i tani mikrokontroler ATtiny2313 odpowiedzialny w zasadzie wyłącznie za obsługę wbudowanego transceivera domyślnie pracującego w trybie odbiornika oraz realizację fazowego sterowania źródłem światła 230 V AC. Jako że nie było potrzeby wyposażania tego układu w jakikolwiek interfejs użytkownika, do konfigurowania adresu *Slave* modułu przewidziano typowy 16-pozycyjny nastawnik DIP-switch, zaś do sygnalizacji faktu odebrania poprawnego rozkazu oraz jego wykonania przewidziano diodę świecącą RX. Zasada działania tego układu jest nieskomplikowana: moduł ten pracujący domyślnie w trybie odbiornika oczekuje na ważną dla niego ramkę danych sterujących (zawierającą jego adres sprzętowy) przesyłaną przez moduł sterujący (*Master*). Po otrzymaniu takiej ramki danych, układ *Slave* przełącza transceiver w tryb nadajnika, następnie odsyła natychmiast potwierdzenie w postaci tejże ramki danych lub danych oczekiwanych przez układ *Master* (w zależności od typu rozkazu sterującego) po czym przełącza transceiver ponownie



Rysunek 4. Graf funkcjonalny algorytmu obsługi układu Master systemu radioDimmer.

Listing 2 Treść procedury odpowiedzialnej za komunikację Master/Slave

```
uint8_t Send_to_node(uint8_t Address, uint8_t Command, uint8_t Data) //Gdy
wynik działania funkcji != Disconected to znaczy, że odebrano poprawną ramkę
danych
{
  uint32_t SentFrame;
  register uint8_t Trials = 0;
//Result != Disconected (0xFF) oznacza, że odebrano poprawną ramkę
  register uint8_t Result = Disconnected;
//Tworzymy kompletną ramkę danych:
//Header | A3-A2-A1-A0-C3-C2-C1-C0 | Data
  SentFrame = (uint32_t) Header << 16;
  SentFrame |= (uint32_t) Address << 12;
  SentFrame |= (uint32_t) Command << 8;
  SentFrame |= (uint32_t) Data;
  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<<TICIE1);
    Tx_to_rx();  //Przełączamy transceiver w tryb odbioru  - 300us
//Odczekujemy przez pewien czas, by Slave
//odpowiedział nam na zapytanie
    _delay_ms(12);
    TIMSK &= ~(1<<TICIE1);  //Dezaktywujemy przerwanie ICP
//Po tym czasie transceiver przełączamy z powrotem
//do nadawania bo to jego stan domyślny - 405us
    Rx_to_tx();
//Układ Slave odpowiedział, a odpowiedź umieszczona
//jest w zmiennej Frame
    if (Frame_ready)
    {
//Sprawdzamy poprawność odebranych danych
//w zależności o typu komendy, która była wysłana
      switch (Command)
      {
        case Set_phase:
//Slave odpowie Dummy_byte jesli zdekoduje błędną komendę
          if (Frame != Dummy_byte) Result = Frame;
          break;
        case Ask_for_phase:
//Slave odpowie Dummy_byte jesli zdekoduje błędną komendę
          if (Frame != Dummy_byte) Result = Frame;
          break;
        }
        Frame_ready = 0;
    }
    Trials++;
    _delay_us(600);
  }
  RESET_ENABLE;  //Przechodzimy do trybu PoweDown
//Zerujemy chociaż nie przechodzimy do RX, ale by mieć
//stan wyjsciowy portów sterujących
  RESET_TX_RX;
  return Result;
}
```

w tryb odbiornika i aktualizuje wartości odpowiednich zmiennych odpowiedzialnych za realizację algorytmu sterowania fazowego.

Rysunek 5. Wygląd głównego ekranu modułu Master.

Rysunek 6. Wygląd ekranu modułu Master dla trybu wyboru układu Slave w celu edycji jego parametrów.

Rysunek 7. Wygląd ekranu modułu Master dla trybu regulacji jasności wybranego układu Slave.

Rysunek 8. Wygląd ekranu modułu Master dla trybu konfiguracyjnego (Setup).

Wspomniane sterowanie wykonano w sposób dość standardowy acz ciekawy w związku, z czym warto zatrzymać się choć na chwilę przy tym zagadnieniu. Zastosowane tzw. sterowanie fazowe, polega na ograniczeniu prądu płynącego przez odbiornik zasilany napięciem sieciowym 230 V AC (w naszym przypadku żarówkę) poprzez „wycięcie" części przebiegu napięcia zasilającego w każdym okresie takiego przebiegu (podobnie jak to ma miejsce przy sterowaniu PWM tyle, że tutaj regulujemy „wypełnienie" sinusoidy napięcia zasilającego). „Wypełnienie" to regulujemy poprzez zmianę czasu, jaki upływa od przejścia przebiegu napięcia zasilającego przez zero do czasu załączenia triaka wykonawczego (im większe jest to opóźnienie tym mniejsza średnia moc dostarczana do odbiornika). Aby tego dokonać niezbędny jest dokładny układ synchronizujący nasz algorytm z przebiegiem napięcia sieciowego. Do tego celu wykorzystano popularny optoizolator PC814, który w swojej strukturze zawiera dwie diody LED połączone przeciwsobnie i fototranzystor wyjściowy. Dzięki temu na wyjściu tego elementu (kolektor wbudowanego fototranzystora) otrzymujemy bardzo wąską szpilkę dla każdego przejścia przez zero przebiegu napięcia sieciowego (dokładnie, na chwile przed przejściem napięcia sieciowego przez zero). Narastające zbocze tej szpilki wyzwala przerwanie zewnętrzne INT0, którego procedura obsługi odpowiedzialna jest za synchronizację regulacji fazowej. W procedurze tej uruchamiany jest z kolei układ czasowo-licznikowy Timer2 w trybie CTC, dla którego parametry pracy dobrano w taki sposób by w czasie jednej połówki przebiegu napięcia zasilającego (pomniejszonej o połowę szerokości „szpilki" wyzwalającej) nastąpiło 50 przerwań OC2 Timera2 (od porównania zawartości licznika z wartością rejestru OCR2). Przerwanie OC2 realizuje natomiast obsługę zaimplementowanej regulacji fazowej poprzez cykliczne załączanie triaka T1 w chwili zależnej od wartości zmiennej globalnej definiującej ustawienia żądanej jasności.

Montaż

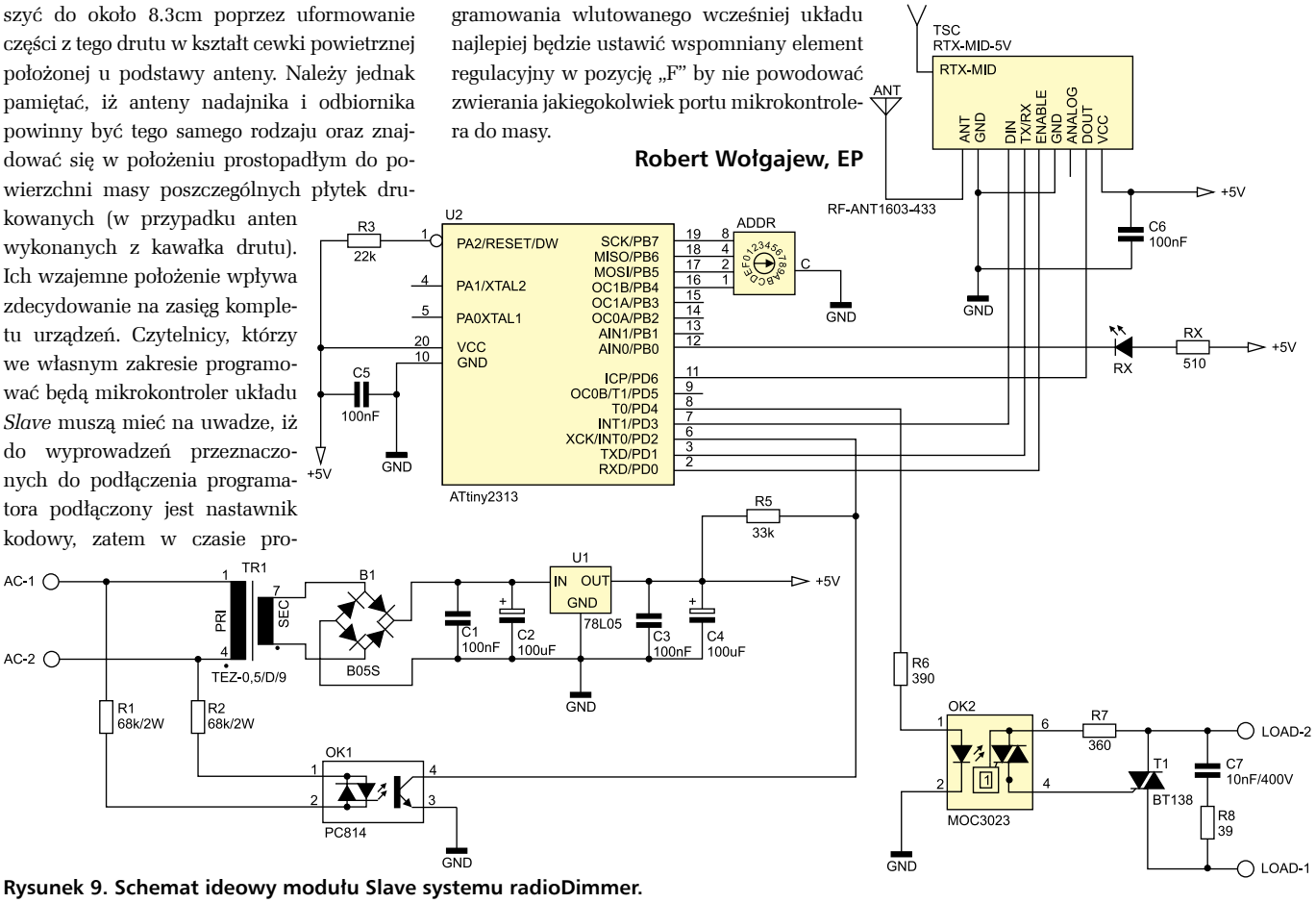
Na **rysunku 10** i **rysunku 11** pokazano wygląd obwodów drukowanych modułów *Master* i *Slave* systemu radioDimmer z zaznaczeniem montażu poszczególnych elementów. Montaż obu modułów zostanie opisany łącznie, gdyż odnosi się do zwyczajowych zasad montażu układów tego typu.

Jak zwykle, należy rozpocząć od wlutowania elementów biernych, następne lutujemy złącza oraz pozostałe elementy mechaniczne, a na końcu półprzewodniki. W wypadku wykorzystania wyświetlacza OLED jako elementu interfejsu użytkownika nie ma potrzeby montażu elementów tranzystora T1, potencjometru P1 oraz rezystorów R4 i R5, gdyż nie korzysta on z mechanizmu wygaszania podświetlenia. Należy zwrócić szczególną uwagę na konieczność pocynowania grubą warstwą cyny ścieżek przewodzących duże prądy tj. ścieżek pomiędzy triakiem T1, a złączem śrubowym LOAD modułu *Slave* (dla tych ścieżek nie stosowano tzw. solder-maski).

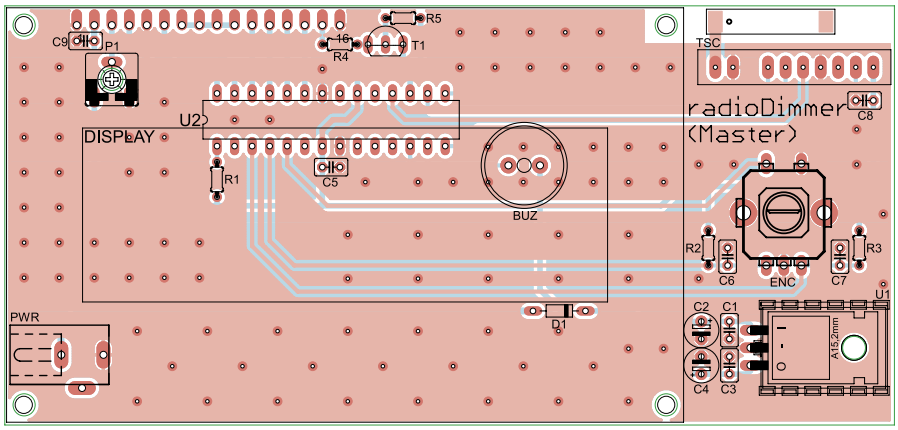
Wyświetlacz OLED należy zamocować w odpowiedniej odległości od obwodu drukowanego modułu Master najlepiej przy pomocy tulei dystansowych wykorzystując przewidziane w tym celu otwory zaś same połączenie należy wykonać przy użyciu listwy goldpin (gniazdo-wtyk) lub zwykłej taśmy wieloprzewodowej. Jako antenę w obu przypadkach przewidziano scaloną antenę wielowarstwową typu SMD, jednak z powodzeniem możemy użyć zwykłego kawałka drutu miedzianego o minimalnym przekroju 0,5 mm^2 i długości około 17,2 cm. Co ciekawe, testy praktyczne pokazały, iż czasami zastosowanie zwykłego kawałka drutu daje lepszy efekt, aniżeli opcjonalna antena SMD (oczywiście zależy to w dużym stopniu od projektu samej płytki drukowanej). Długość takiej anteny można zmniej-

szyć do około 8.3cm poprzez uformowanie części z tego drutu w kształt 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 masy poszczególnych płytek drukowanych (w przypadku anten wykonanych z kawałka drutu). Ich wzajemne położenie wpływa zdecydowanie na zasięg kompletu urządzeń. Czytelnicy, którzy we własnym zakresie programować będą mikrokontroler układu *Slave* muszą mieć na uwadze, iż do wyprowadzeń przeznaczonych do podłączenia programatora podłączony jest nastawnik kodowy, zatem w czasie programowania wlutowanego wcześniej układu najlepiej będzie ustawić wspomniany element regulacyjny w pozycję „F" by nie powodować zwierania jakiegokolwiek portu mikrokontrolera do masy.

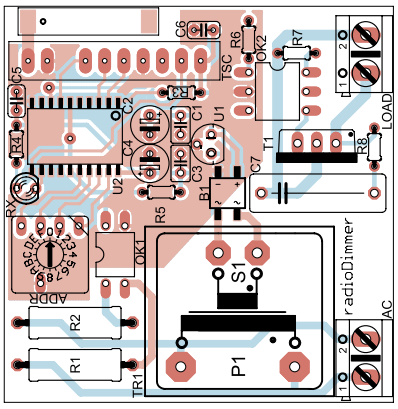
Robert Wołgajew, EP



Rysunek 9. Schemat ideowy modułu Slave systemu radioDimmer.



Rysunek 10. Schemat montażowy modułu Master systemu radioDimmer.



Rysunek 11. Schemat montażowy modułu Slave systemu radioDimmer.