



Zdalne sterowanie z dowolnym pilotem RC5



W jednym z poprzednich numerów *Elektroniki dla Wszystkich* pozwoliłem sobie zaprezentować Czytelnikom opis budowy uniwersalnego pilota RC5. Wspomniałem także, że wykonanie tego urządzenia ma sens, ponieważ w najbliższym czasie zaprezentuję Wam szereg urządzeń sterowanych z tego właśnie pilota. Jak dotąd dotrzymałem obietnicy tylko częściowo, opracowując dla Was prostą zabawkę sterowaną kodem RC5. Najwyższy więc czas, aby kontynuować rozpoczętą pracę i zaferować Czytelnikom *Elektroniki dla Wszystkich* kolejne urządzenie, które jednak tym razem bynajmniej nie będzie tylko zabawką. Zanim jednak przejdziemy do opisu proponowanego układu, chciałbym wyjaśnić pewne nieporozumienia i rozproszyć wątpliwości, które nękają część z Was.

Jednym z pierwszych tematów poruszonych w ramach wykładów BASCOM College było odbieranie i dekodowanie transmisji kodu RC5 za pomocą procesorów typu AT89C2051. Temat ten spotkał się ze sporym zainteresowaniem studentów BASCOM College, którzy jednak napotkali na pewne trudności podczas realizacji związanych z nim ćwiczeń. Trudno mi nawet policzyć, ile otrzymałem rozpaczliwych maili, w których pisałeś, że opisywane przeze mnie polecenia w ogóle nie działają, że na ekranie wyświetlacza LCD pokazują się same bzdury i że ten cały BASCOM to w ogóle jest do kitu. Niestety, ci, którzy powątpiewali w poprawność zawartych w języku MCS BASIC poleceń, absolutnie nie mieli racji. Powód ich, niepowodzeń był prawie zawsze taki sam: do eksperymentów używali oni pilotów od sprzętu RTV, zapominając, że nie każdy taki pilot musi pracować w standardzie RC5. Wystarczy tylko zajrzeć do dowolnego sklepu ze sprzętem radiowo-telewizyjnym, aby przekonać

się, że nawet większość pilotów używanych w Polsce nie obsługuje tego rodzaju transmisji. Firmy japońskie i dalekowschodnie produkujące sprzęt RTV zyskały sobie znaczącą pozycję na naszym rynku, a właśnie ci producenci nie stosują w swoich wyrobach kodu RC5, będącego standardem w sprzęcie produkowanym w Europie (np. w sprzęcie produkowanym przez Philipsa).

O tym, co napisałem musicie pamiętać „przymierzając się” do budowy opisanego niżej urządzenia. Jeżeli posiadacie w domu wyłącznie sprzęt elektroniczny produkcji japońskiej lub innej firmy z Dalekiego Wschodu, to wchodzące w jego skład piloty okażą się zupełnie bezużyteczne przy współpracy z naszym układem. Będziecie wtedy mieli różne drogi wyjścia z kłopotliwej sytuacji. Będziecie mogli zakupić jakiegoś pilota, upewniając się przedtem, że na pewno pracuje on z kodem RC5. Jednak takie rozwiązanie z pewnością nie zadowoli fanów elektroniki, jakimi jesteście i pozostanie Wam wykonanie pilota opisanego w EdW albo też uniwersalnego nadajnika kodu RC5, umożliwiającego wysłanie dowolnej komendy RC5 pod dowolnie wybrany adres, którego opis został zamieszczony w numerze 1/00 *Elektroniki Praktycznej* (AVT-849).

Z pozoru proponowany układ nie różni się zbyt wiele od wielu mu podobnych. Umożliwia sterowanie ośmioma odbiornikami energii elektrycznej, a ze względu na zastosowanie przekaźników mocy jako elementów wykonawczych nie ma większego znaczenia czy będą to układy zasilane prądem przemiennym z sieci energetycznej, czy też wymagające dostarczenia prądu stałego. Takich układów zbudowano wiele, ale ten, z którym zapoznamy się dzisiaj ma pewną szczególną cechę, która, jak mam nadzieję, znajdzie uznanie

Czytelników *Elektroniki dla Wszystkich*. Proponowany układ nie jest związany za jakimkolwiek konkretnym pilotem i może współpracować z dowolnym nadajnikiem kodu RC5, w tym oczywiście z pilotem AVT-2427. Nie ma najmniejszego znaczenia, jakie komendy wykorzystamy do sterowania naszym układem. Mogą to być dowolne polecenia z 64 komend, które zawiera standard RC5, wysyłane pod całkowicie dowolny spośród 32 dostępnych adresów. Nasz układ podczas pierwszego uruchomienia musi zostać „nauczony”, jakie komendy będziemy wykorzystywać i pod jaki adres będą one wysyłane. Zapamiętane informacje przechowywane są w pamięci EEPROM i mogą zostać stamtąd usunięte lub zmienione jedynie w wyniku naszego celowego działania.

Jak się za chwilę przekonamy, sercem układu i jego jedynym aktywnym elementem jest procesor typu AT89C2051, niezwykle popularny zarówno wśród hobbystów, jak i profesjonalnych konstruktorów, „gwiazda” naszego kursu BASCOM College. Program sterujący pracą procesora został napisany, przetestowany i skompilowany za pomocą narzędzi zawartych w pakiecie BASCOM8051 „Special Edition for Elektronika Praktyczna”, który przebojem zyskuje sobie coraz większą popularność wśród Czytelników naszego pisma, podobnie jak wśród elektroników na całym świecie. Dlatego też opis działania układu ilustrowany będzie licznymi fragmentami programu napisanego w języku MCS BASIC.

Wykonanie proponowanego układu jest stosunkowo proste, a koszt elementów potrzebnych do jego budowy jest niezbyt wysoki. Koledzy, którzy zdecydują się na jego budowę, będą mieli dwie możliwości zaopatrzenia się w zaprogramowany procesor, niezbędny do funkcjonowania urządzenia.

Metodą najprostszą będzie zakupienie kitu, w którego skład wchodzi zaprogramowany chip AT89C2051. Bardziej ambitni konstruktorzy zechcą zapewne samodzielnie zaprogramować procesor, pisząc własną wersję programu. W tym celu powinni się zaopatrzyć w podstawowe narzędzia jakimi są pakiet BASCOM8051 i programator MCS Flashprogrammer AVT-2502. Dostęp do programu BASCOM8051 Special Edition For Elektronika Praktyczna jest szczególnie łatwy: znajduje się na płycie CD-EP 4/00 oraz na stronie www.ep.com.pl.

Jak to działa?

Na rysunku 1 został pokazany schemat elektryczny proponowanego układu sterownika. Układ jest wyjątkowo prosty: poza procesorem 89C2051 i pamięcią AT24C04 układ składa się z ośmiu przełączników wykonaw-

czych, odbiornika kodu RC5 - TFMS5360 i garstki elementów dyskretnych. Procesor 89C2051 posiada liczne zalety, ale także i jedną wadę: nie jest wyposażony w wewnętrzną nieulotną pamięć danych, co spowodowało konieczność zastosowania dodatkowego układu, przechowującego informacje o kodach poszczególnych komend sterujących pracą urządzenia.

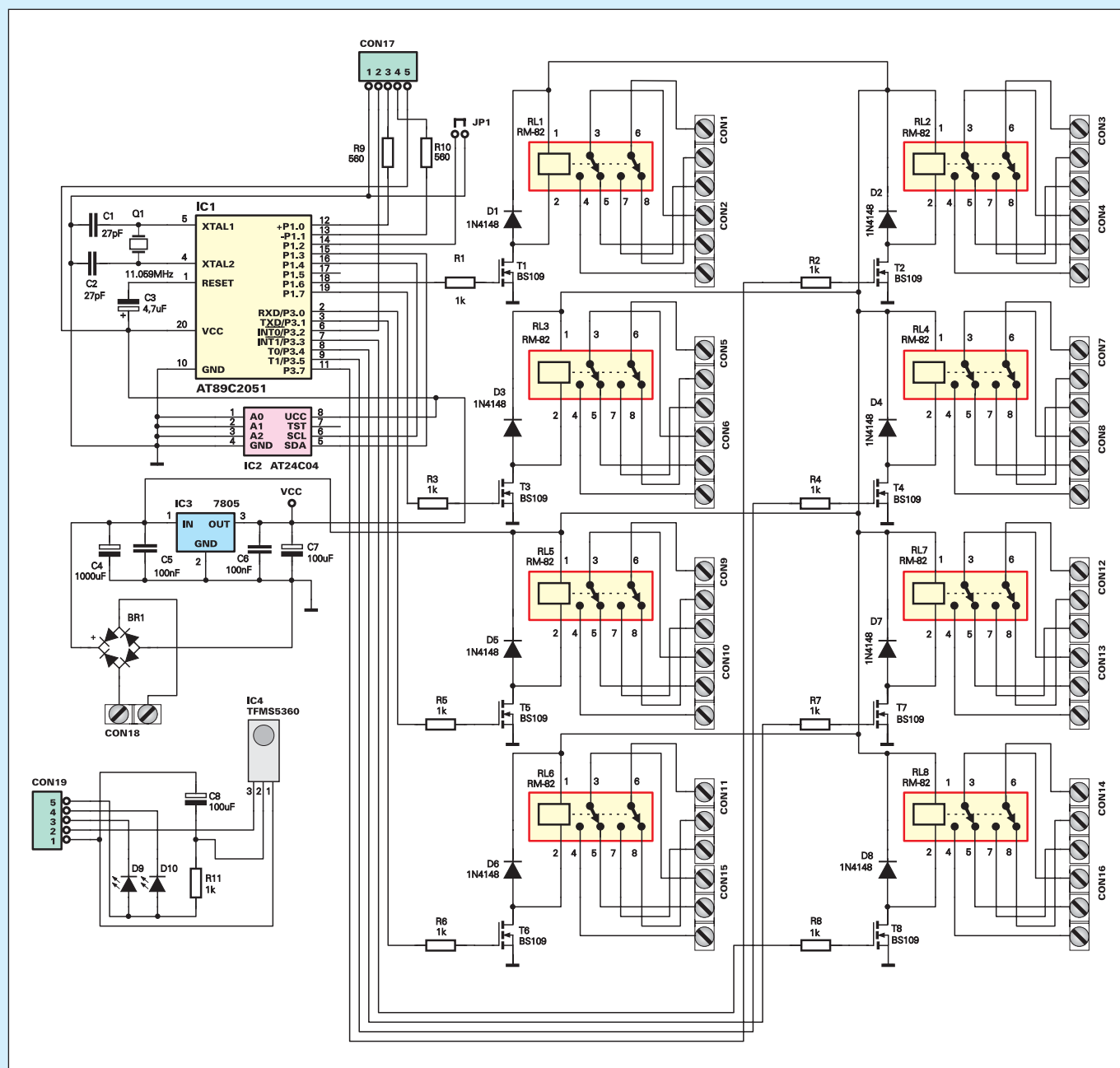
Jako elementy wykonawcze zastosowane zostały przełączniki typu RM-82, z których każdy wyposażony jest w dwie pary przełączanych styków. Zwiększa to dodatkowo możliwości układu, który w zasadzie może sterować aż 32 różnymi urządzeniami, dołączanymi w najrozmaitszy sposób do styków przełączników. Szerzej poruszymy ten temat w części artykułu poświęconej montażowi układu.

Jako odbiornik kodu RC5 emitowanego przez pilota zastosowano dobrze znany układ typu TFMS5360. Jest to „magiczna” kostka, która uwalnia nas od wszelkich problemów związanych z budową wzmacniacza odbieranego sygnału czy też filtrów eliminujących „niepożądane” częstotliwości nośne. Na wyjściu IC4 otrzymujemy od razu „gotowy” przebieg prostokątny, zgodny ze standardem TTL.

Układ może być zasilany napięciem stałym o wartości 10 ... 18VDC lub przemianym 8 ... 12VAC dołączanym do złącza CON19. Część cyfrowa urządzenia zasilana jest napięciem +5VDC, dostarczanym przez scalony stabilizator napięcia IC3.

Sposób połączenia procesora z biernymi elementami układu najlepiej prześledzić na podstawie fragmentu programu, w którym

Rys. 1 Schemat ideowy



za pomocą wyjątkowo wygodnego polecenia ALIAS nadano wypro-
wadzeniom procesora nazwy zgodne z pełno-
nymi przez nie funkcjami. Dodatkowo w syste-
mie utworzono magi-
stralę I²C, która służy
obsłudze zewnętrznej
pamięci danych - I²C.

Analizę sposobu
działania układu roz-
poczniemy opierając się na listingach frag-
mentów programu w momencie włączenia
zasilania urządzenia. Pierwszą czynnością ja-
ką procesor wykonuje po inicjalizacji i konfi-
guracji systemu jest sprawdzenie stanu jum-
pera JP1 (listing 1). Od wyniku tego badania
zależać będzie dalszy bieg wypadków.

```
Config Sda = P1.3
Config Scl = P1.4
```

```
R11 Alias P1.6
R12 Alias P3.7
R13 Alias P1.7
R14 Alias P3.5
R15 Alias P3.0
R16 Alias P3.1
R17 Alias P3.4
R18 Alias P3.3
Jp1 Alias P1.2
Led1 Alias P1.1
Led2 Alias P1.0
```

Zauważmy, że niezależnie od badania stu-
nu jumpera program odczytuje także zawar-
tość komórki pamięci o adresie 127 i jeżeli nie
jest ona równa 133, to także wzywa podpro-
gram rejestracji komend. Za chwilę dowiemy
się, jaki sens ma ten fragment programu.

Niezależnie, czy zostało to spowodowane
zwarcieniem jumpera JP1, czy też wykryciem
nieprawidłowej wartości zapisanej pod adre-
sem 133 pamięci EEPROM, program rozpo-
czynia wykonywanie pętli zawartej w podpro-
gramie REGISTRATION, pokazanym na lis-
tingu 2. Jego zadaniem jest teraz zarejestro-
wanie 8 komend kodu RC5, z których każda
może być wysłana pod dowolny z 32 adresów.

Wykonanie tego programu spowoduje zapi-
sanie w pamięci EEPROM 8 komend i ośmiu
adresów wyemitowanych z pilota RC5 i prze-
chowanie ich tam w celu późniejszego porów-
niania z sygnałami odbieranymi podczas pracy
układu. Wiemy już teraz, dlaczego nasz pro-
gram sprawdza zawartość 127 komórki pamię-
ci EEPROM: podczas rejestracji poleceń jest
tam zapisywana liczba 133, a inna wartość od-
czytana z tej komórki świadczy, że pamięć nie
była jeszcze programowana w naszym ukła-

dzie. Oczywiście wartość 133 jest całkowicie
przypadkowa, po prostu jest to pierwsza liczb-
ba, jak wpadła mi do głowy.

Zwróćmy jeszcze uwagę na jedną cechę
opisywanego układu. W pamięci zapisywane są
nie tylko polecenia, ale i adresy pod jakie zosta-
ły wysłane. Z tego wynika, że do obsługi nasze-
go układu możemy z powodzeniem wykorzy-
stywać różne piloty, przeznaczone do sterowa-
nia różnymi urządzeniami. Mogą to być piloty
od telewizora, magnetowidu, radiomagnetofonu
czy też od kamery wideo. Ważne jest tylko
to, aby wszystkie pracowały w kodzie RC5.

Po zarejestrowaniu poleceń sterujących
układ rozpoczyna pracę w pętli (listing 3),
w której oczekuje na odebranie jakiejś trans-
misji nadawanej w kodzie RC5. Warto zau-
ważyć, że wyjście odbiornika podczerwieni
IC3 zostało dołączone za pośrednictwem złą-
cza CON17 + CON19 do wejścia przerwania
INT0 procesora. Jeżeli więc do odbiornika
dotrze jakikolwiek sygnał nadawany na pod-
czerwieni z częstotliwością 36kHz, to jego
wyjście przybierze stan niski, a procesor
przystąpi do realizacji podprogramu obsługi
przerwania, pokazanej na listingu 4.

```
'Listing 1
Set Jp1                'spróbuj ustawić
                       stan wysoki na pinie Jp1
If Jp1 = 0 Then        'jeżeli próba
                       nieudana, to:
Call Registration      'wezwiń podprogram
                       rejestracji komend i adresów
End If                 'koniec warunku
Call Read_eeprom(127, Value) 'odczytaj zawartość
                       komórki 127 pamięci EEPROM
If Value <> 133 Then    'jeżeli odczytana
                       wartość nie jest równa 133, to
Call Registration      'wezwiń podprogram
                       rejestracji komend i adresów
End If                 'koniec warunku
```

```
'Listing 4
Receiver5:
Disable Int0          'chwilowo zawieś obsługę przerwania INT0
Getrc5(subaddress, Command) 'zbadaj, czy odebrany kod to transmisja RC5 i dokonaj jego analizy
If Command < 63 Then 'jeżeli wartość odebranej komendy jest mniejsza od 63, to:
If Subaddress < 31 Then 'jeżeli wartość adresu, pod który wysłana została komenda
                       jest mniejsza od 31, to:
Rc5_received = 1      'zmienna pomocnicza RC5_RECEIVED sygnalizująca odebranie poprawnej
                       transmisji przyjmuje wartość jeden
End If : End If       'koniec warunków
Return
```

```
'Listing 2
Sub Registration
Disable Int0          'chwilowo zawieś obsługę przerwania INT0
Eepromadres = 1      'zmienna EEPROMADRES przyjmuje wartość 1
Temp2 = 1            'zmienna pomocnicza TEMP2 przyjmuje wartość 1

For Temp = 1 To 10 'dziesięciokrotnie:
Call Redshort        'wezwiń podprogram REDSHORT, którego zadaniem jest generowanie krótkiego błysku 'czerwonej
                       diody LED. Błyski tej diody informują użytkownika, że może przystąpić do rejestracji poleceń
Next

Enable Int0          'ponownie uaktywnij obsługę przerwania INT0
Do                  'początek pętli programowej
If Rc5_received = 1 Then 'jeżeli zmienna pomocnicza RC5_RECEIVED przyjęła wartość jeden, to:
Disable Int0          'zawieś obsługę przerwania INT0
Rc_received = 0      'przywróć zmiennej RC5_RECEIVED wartość zero
Call Write_eeprom(eepromadres, Subaddress) 'zapisz w pamięci EEPROM adres, pod jaki 'wysłane zostało
                       polecenie nadane z pilota
Incr Eepromadres      'zwiększ o 1 wartość zmiennej EEPROMADRES
Call Write_eeprom(eepromadres, Command) 'zapisz w pamięci EEPROM odebraną
                       komendę kodu RC5
Incr Eepromadres      'zwiększ o 1 wartość zmiennej EEPROMADRES
Incr Temp2           'zwiększ o 1 wartość zmiennej pomocniczej TEMP2
For Temp = 2 To Temp2 'tylko raz, ile komend zostało odebranych:
Call Greenlong        'wezwiń podprogram generujący długi błysk diody zielonej,
Next Temp
Enable Int0          'ponownie uaktywnij obsługę przerwania INT0
End If

If Temp2 = 9 Then    'jeżeli zmienna TEMP2 przyjęła wartość 9, czyli zapisane już zostało 8 komend i adresów, to:
Call Write_eeprom(127, 133) 'zapisz pod adresem 127 pamięci EEPROM wartość 133
Exit Do              'wyjdź z pętli programowej
End If               'koniec warunku
Loop
End Sub
```

```
'Listing 3
Sub Mainloop          'główna pętla programowa
Do
If Rc5_received = 1 Then 'jeżeli zmienna pomocnicza
                       sygnalizująca odebranie transmisji RC5 przyjęła
                       wartość 1, to:
Rc5_received = 0      'zmienna ta z powrotem
                       przyjmuje wartość 0
Disable Int0          'chwilowo zawieszamy
                       obsługę dalszych przerwania INT0
Call Code_identification
'wezwiń podprogram analizujący odebrany kod
Wait 1                'zaczekaj sekundę
Enable Int0           'powtórnie wznów obsługę
                       przerwania INT0
End If                'koniec warunku
Loop
End Sub
```

Jeżeli wstępna analiza odebranego sygnału
wypadła pomyślnie, czyli że program stwier-
dził, że ma do czynienia z kodem RC5, to we-
zwana zostaje procedura identyfikacji odebra-
nej transmisji. W tym momencie mamy już do
dyspozycji dwie dane: wartość odebranej ko-
mendy i adres, pod jaki został wysłana. Pozo-
staje zatem porównać te dane z wartościami
zapisanymi w pamięci EEPROM. Czynność tę
wykonuje podprogram pokazany na listingu 5.

Omówiliśmy najważniejsze fragmenty programu sterownika. Mam nadzieję, że wszyscy Czytelnicy zrozumieli zasadę jego działania i bez jakichkolwiek wątpliwości mogą już przystąpić do budowy proponowanego układu.

Montaż i uruchomienie

Na **rysunku 2** zostało pokazane rozmieszczenie elementów na dwóch płytkach obwodów drukowanych. Na większej płytce umieszczona została główna część układu,

Jedynie część układu zawierająca odbiornik RC5 musi być zlokalizowana w widocznym miejscu, ale sądzę, że z łatwością znajdziecie jakąś elegancką obudowę dla tej małej płytki.

Montaż układu wykonujemy w typowy sposób, rozpoczynając od wlutowania elementów o najmniejszych gabarytach, a kończąc na kondensatorach elektrolitycznych i przełącznikach. Pod układy scalone należy zastosować podstawki. Przełączniki zastosowane w układzie są najkosztowniejszymi elementami całego urządzenia. Dlatego też, jeżeli nie mamy zamiaru wykorzystywać wszystkich ośmiu kanałów, jakimi dysponujemy, to montowanie ośmiu przełączników nie miałoby większego sensu ekonomicznego. Niejednokrotnie wystarczy nam dwa, trzy kanały, a przełączniki obsługujące pozostałe urządzenia możemy zawsze zamontować później.

Obie płytki należy połączyć ze sobą za pomocą sześciżyłowego przewodu, najlepiej tzw. taśmowego. Układ prototypowy testowany był z przewodem o długości 2,5 mb i nie stwierdzono jakichkolwiek zakłóceń w przekazywaniu informacji z odbiornika RC5 do procesora.

Układ sterownika zmontowany ze sprawdzonych elementów działa natychmiast po włożeniu zaprogramowanego procesora w podstawkę i poza zaprogramowaniem komend sterujących i ewentualnie kodu dostępu nie wymaga jakichkolwiek czynności uruchomieniowych.

Układ powinien być zasilany napięciem stałym o wartości ok. 14VDC lub przemienionym ok. 8VAC.

Dla tych Kolegów, którym nie bardzo chciało się wniknąć w szczegóły budowy programu obsługującego nasz sterownik w największym skrócie podamy teraz „instrukcję obsługi” zbudowanego układu.

Nowo zbudowany, wyposażony w „czystą” pamięć EEPROM układ będzie wymagał zarejestrowania poleceń kodu RC5. W takim układzie nie musimy nawet zwierać jumpera JP1, ponieważ po wykryciu nie zaprogramowanej pamięci EEPROM przejdzie on od razu w tryb rejestracji komend, co zostanie zasygnalizowane błyskami czerwonej diody LED. Następnie pojedynczy błysk diody zielonej stanie się zachętą do podania pierwszej komendy. Po naciśnięciu przycisku pilota, zarejestrowanie pierwszej komendy zostanie potwierdzone dwoma błyskami zielonej diody, co jednocześnie jest zachętą do

```

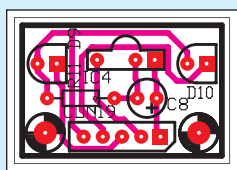
'Listing 5
Sub Code_identification
Eepromadres = 1 'zmienna pomocnicza EEPROMADRES przyjmuje wartość 1
For Temp = 1 To 8 'ośmiokrotnie:
Call Read_eeprom(eepromadres , Subaddress2) 'odczytaj z pamięci EEPROM zapisany tam adres pod który
' wysyłane jest polecenie dla kolejnego kanału

Incr Eepromadres 'zwiększ wartość zmiennej pomocniczej EEPROMADRES o 1
Call Read_eeprom(eepromadres , Command2) 'odczytaj z pamięci EEPROM wartość komendy obsługującej
' kolejny kanał

Incr Eepromadres 'zwiększ wartość zmiennej pomocniczej EEPROMADRES o 1
If Command = Command2 Then 'jeżeli wartość odebranej komendy jest równa zapisanej, to
If Subaddress = Subaddress2 Then 'jeżeli wartość odebranego adresu jest równa odczytanej
' z pamięci, to:

Select Case Temp 'dokonaj analizy zmiennej TEMP
Case 1 : R11 = Not R11 'jeżeli zmienna TEMP = 1 to zmień stan przełącznika RL1 na przeciwny
Case 2 : R12 = Not R12 'jeżeli zmienna TEMP = 1 to zmień stan przełącznika RL1 na przeciwny
Case 3 : R13 = Not R13 'jeżeli zmienna TEMP = 1 to zmień stan przełącznika RL1 na przeciwny
Case 4 : R14 = Not R14 'jeżeli zmienna TEMP = 1 to zmień stan przełącznika RL1 na przeciwny
Case 5 : R15 = Not R15 'jeżeli zmienna TEMP = 1 to zmień stan przełącznika RL1 na przeciwny
Case 6 : R16 = Not R16 'jeżeli zmienna TEMP = 1 to zmień stan przełącznika RL1 na przeciwny
Case 7 : R17 = Not R17 'jeżeli zmienna TEMP = 1 to zmień stan przełącznika RL1 na przeciwny
Case 8 : R18 = Not R18 'jeżeli zmienna TEMP = 1 to zmień stan przełącznika RL1 na przeciwny

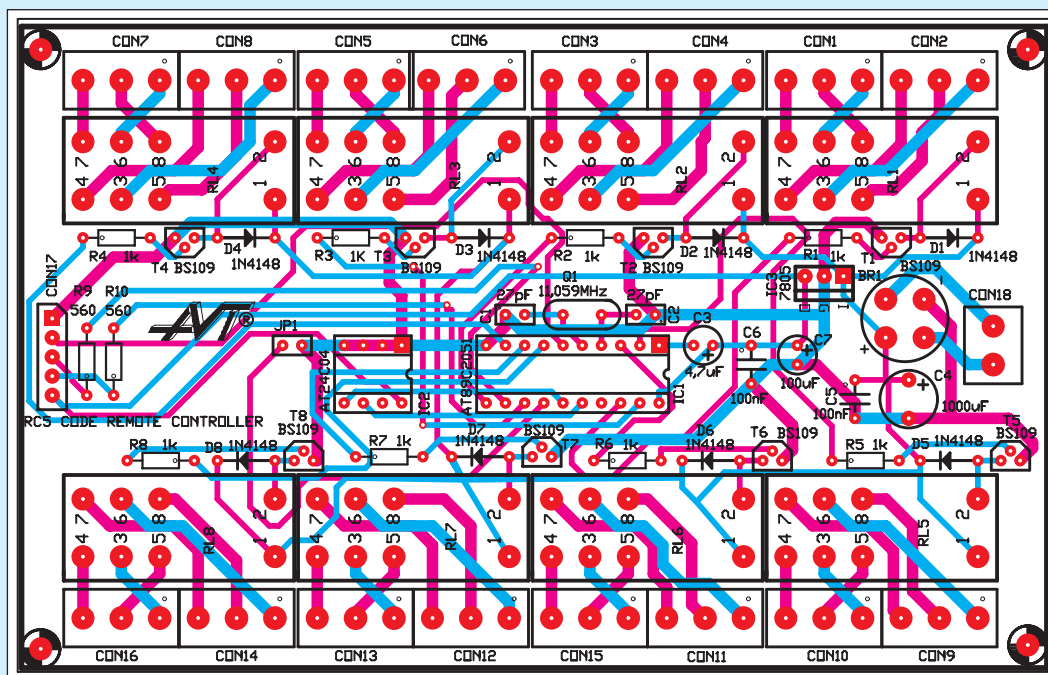
End Select
Call Greenlong 'wygeneruj długi błysk zielonej diody
End If 'koniec warunku
End If 'koniec warunku
Next Temp
End Sub
    
```



Rys. 2a Schemat montażowy

Rys. 2b Schemat montażowy

zawierająca procesor, przełączniki, zasilacz i diody sygnalizujące stan przełączników. Na mniejszej płytce znajduje się tylko odbiornik kodu RC5 i dwie diody sygnalizacyjne. Takie rozmieszczenie elementów i podział układu na dwa moduły umożliwia zamontowanie głównej części urządzenia w obudowie, która niekoniecznie musi spełnić wysokie wymagania estetyczne i spowodować umieszczenie całości pod przysłowiową „szafą”.



podania drugiej komendy, a następnie trzeciej (trzy błyski zielonej diody) i tak dalej, aż do zarejestrowania wszystkich ośmiu poleceń. Jest całkowicie obojętne, które klawisze wykorzystamy do przesyłania poleceń do sterownika, tak jak obojętne jest pod jaki adres będą wysyłane.

Zarejestrowanie wszystkich ośmiu poleceń zostanie przez sterownik skwitowane włączeniem zielonej diody LED na 3 sekundy. Następnie układ przechodzi w stan oczekiwania na polecenia, w którym pozostanie aż do ewentualnego wyłączenia zasilania. Po zaprogramowaniu poleceń należy zawsze pamiętać o usunięciu jumpera JP1, ponieważ w przeciwnym przypadku nasz układ

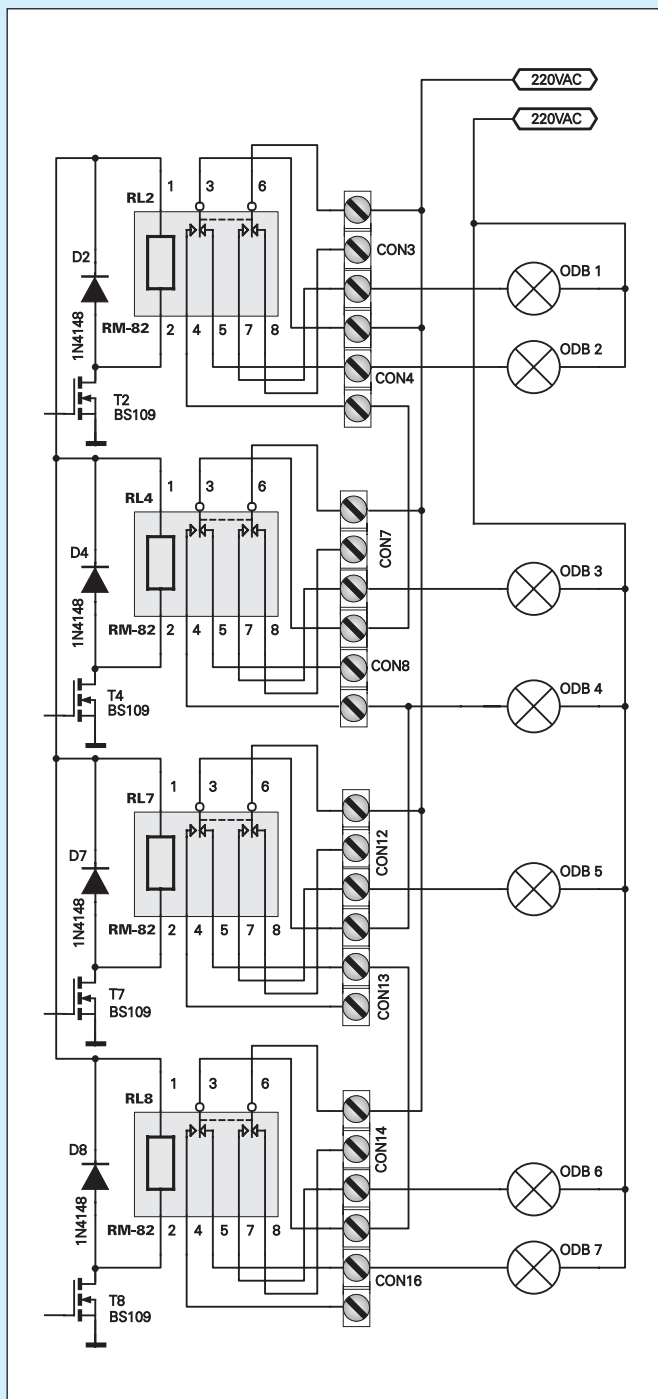
wchodziłby w tryb programowania komend po każdym wyłączeniu zasilania (nawet przypadkowym).

Warto teraz jeszcze wspomnieć o dodatkowych możliwościach oferowanych przez nasz układ, których istnienie zostało już wcześniej zasygnalizowane. Mam tu na myśli wykorzystywanie podwójnych styków przekaźników do włączania i wyłączania dodatkowych urządzeń. Najlepiej posłużmy się tu przykładem, pokazanym na **rysunku 3**. Mamy tam siedem różnych odbiorników

energii elektrycznej dołączonych do czterech z ośmiu przekaźników naszego sterownika. Waszym zadaniem będzie teraz przeanalizowanie, w jaki sposób działa układ zmontowany, a właściwie okablowany zgodnie z tym rysunkiem.

Zbigniew Raabe

e-mail: zbigniew.raabe@edw.com.pl



Wykaz elementów

Kondensatory

C1, C2	27pF
C3	4,7μF/16
C4	1000μF/16
C5, C6	100nF
C7, C8	100μF/16

Rezystory

R1... R8	1kΩ
R9, R10	560Ω
R11	1kΩ

Półprzewodniki

D1... D8	1N4148
D9, D10	LED

IC1	AT89C2051
IC2	AT24C04
IC37805
IC4TFMS5360
T1 ... T8BS109

Pozostałe

BR1	mostek prostowniczy 1A
CON1 ... CON16ARK3
CON175x goldpin
CON18ARK2
CON19	złącze szufladkowe 5
JP12x goldpin + jumper
Q1	rezonator kwarcowy 11.059MHz
RL1 ... RL8RM-82

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2480

R E K L A M A R E K L A M A R E K L A M A



Hurtownia części elektronicznych

Firma Piekarz S.C.
ul. Woluhen 53 paw.66 01-912 Warszawa
tel./fax (022)663-76-01 0-502-270-642
tel./fax (022)835-84-91 835-85-82

Sklep nr 3: teren WGE, pawilon 15
róg al. Niepodległości i Armii Ludowej
tel. (022)825-91-00 wew. 119

- ✓ sprzedaż hurtowa i detaliczna
- ✓ sprzedaż wysyłkowa
- ✓ komplekacja dostaw
- ✓ przyjmujemy zapytania o towary, których nie posiadamy w ofercie
- ✓ nowości: import z firmy Highly Electric z Tajwanu - przyciski, mikroprzełączniki, przełączniki, stacyjki i inne

Cennik: www.piekarz.pl
Zamówienia: firma@piekarz.pl