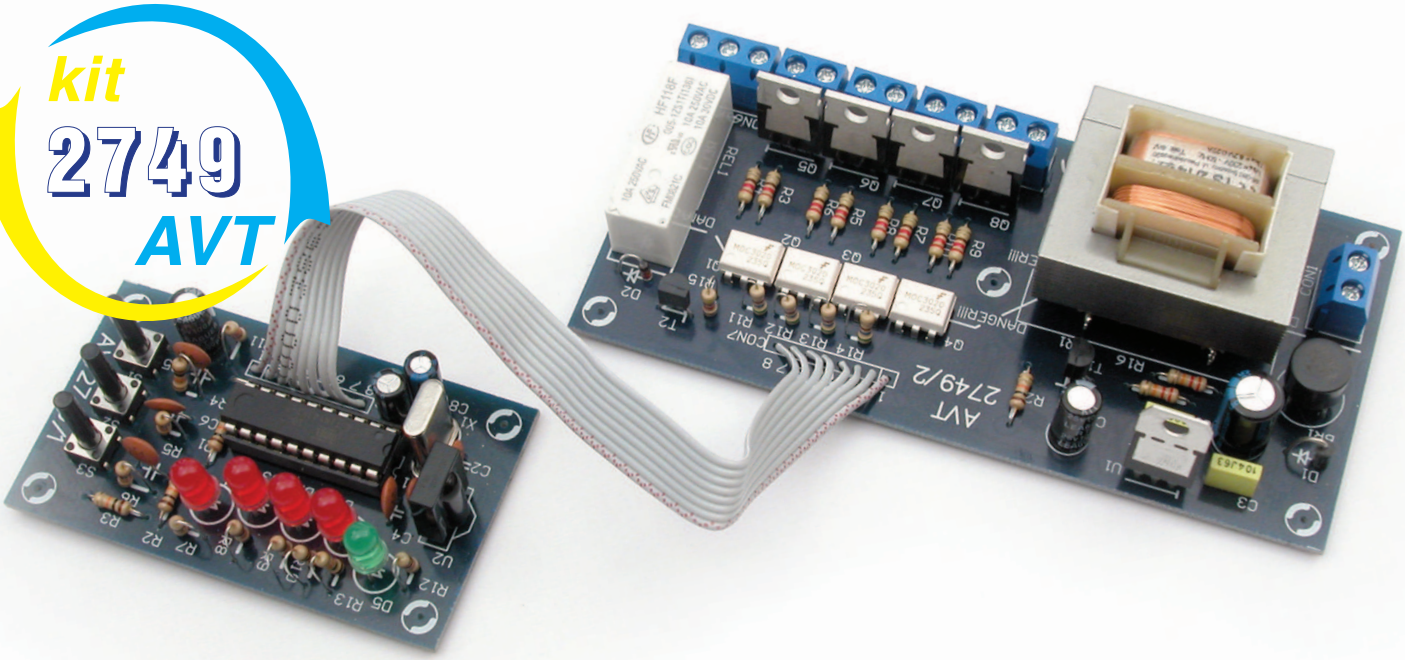
kit 2749 AVT



Czterokanałowy regulator oświetlenia

Na łamach EdW ukazało się już kilka regulatorów jasności żarówek. Niektóre były zbudowane z użyciem „zwykłych" elementów, inne, jak ten opisany w niniejszym artykule, oparte były o mikroprocesory. Każdy z nich miał swoje zalety i wady. Budując ten regulator, starałem się oczywiście, by miał jak najwięcej zalet, czyli maksymalną liczbę funkcji przy minimalnej komplikacji układu. Na ile udało mi się to osiągnąć – zapraszam do dalszej lektury.

Charakterystyka i funkcje tego układu wyglądają następująco:

- Płynne i niezależne sterowanie jasnością czterech lamp.
- Włączanie/wyłączanie naraz wszystkich czterech lamp.
- Zwiększanie/zmniejszanie jasności wszystkich lamp jednocześnie.
- Dodatkowy przekaźnik do sterowania włącz/wyłącz głównego oświetlenia. W tym obwodzie można stosować popularne energooszczędne świetlówki kompaktowe.
- Współpraca z dowolnym pilotem RC-5. Urządzenie „uczy się" komend dowolnych przycisków pilota.
- Możliwość zapisania dziesięciu własnych profili jasności świecenia. Jasność każdej żarówki zapisywana jest oddzielnie.
- Obsługa urządzenia możliwa jest także poprzez trójprzyciskową klawiaturę.
- O stanie, w jakim znajduje się urządzenie, informuje nas pięć diod LED.
- Wszystkie profile, a także adres i komendy pilota przechowywane są w pamięci EEPROM. Oznacza to, że po zaniku napięcia w sieci nasze profile nie znikną oraz to, że wystarczy tylko raz zarejestrować pilota.
- Zarezerwowano także dodatkowy bajt w pamięci EEPROM, który przechowuje informację o stanie przekaźnika sterującego oświetleniem głównym. Potrzebę przechowywania takiej informacji wyjaśnię w dalszej części artykułu.
- Urządzenie posiada także funkcję automatycznego ściemniania lamp - Timer Fader. Po jej włączeniu lampy(a) zaczynają powoli gasnąć od zadanej jasności do całkowitego zaciemnienia. Funkcja ta pomoże spokojnie zasnąć np. małemu dziecku. Czas wygaszania lamp zależy od początkowej jasności. Dla maksymalnej jasności początkowej wynosi około 20min. Czas ten można w bardzo prosty sposób zmienić poprzez zmianę jednej linijki w programie.

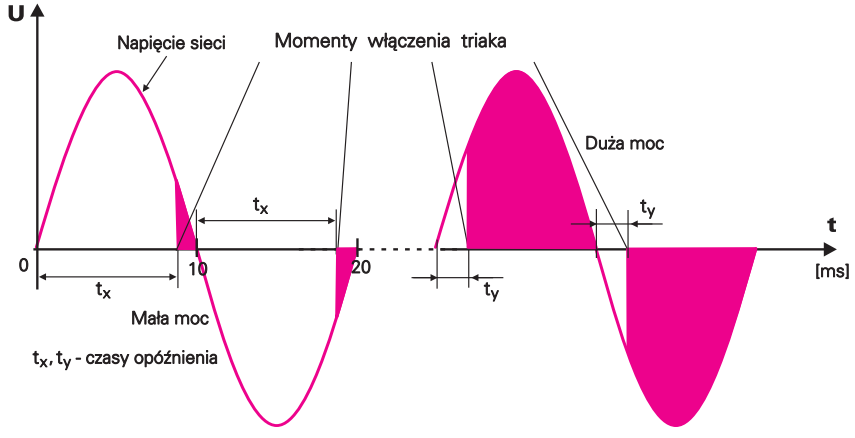
Urządzenie w moim pokoju steruje żyrandolem na suficie oraz czterema lampkami umieszczonymi w rogach pokoju. Daje to mnóstwo możliwości naświetlenia, a dzięki profilom można natychmiastowo zmieniać wymagany rodzaj oświetlenia.

Program napisany został w BASCOM-AVR i zajmuje całą pamięć użytego mikroprocesora (2048B). Kod źródłowy i plik wsadowy można ściągnąć ze strony internetowej EdW z działu Download/Programy do projektów.

Opis układu

Układ reguluje moc dostarczaną do żarówek za pomocą sterowania fazowego. W tym przypadku redukcja mocy polega na opóźnionym włączaniu triaka po przejściu napięcia sieci przez zero. Im opóźnienie to jest większe, tym mniejsza moc dostarczana jest do żarówek. Ideę takiej regulacji przedstawia **rysunek 1**. Jasne pola pod sinusoidą to czas, gdy żarówka nie świeci, czerwone – gdy świeci. Taka regulacja przypomina regulację PWM, tylko tu dodatkowo trzeba jeszcze synchronizować działanie układu z chwilowym napięciem

Rys. 1 Fazowa regulacja mocy



w sieci. Wszystko dlatego, że napięcie w sieci jest sinusoidą, gdyby było np. falą prostokątną, taka synchronizacja byłaby zbędna.

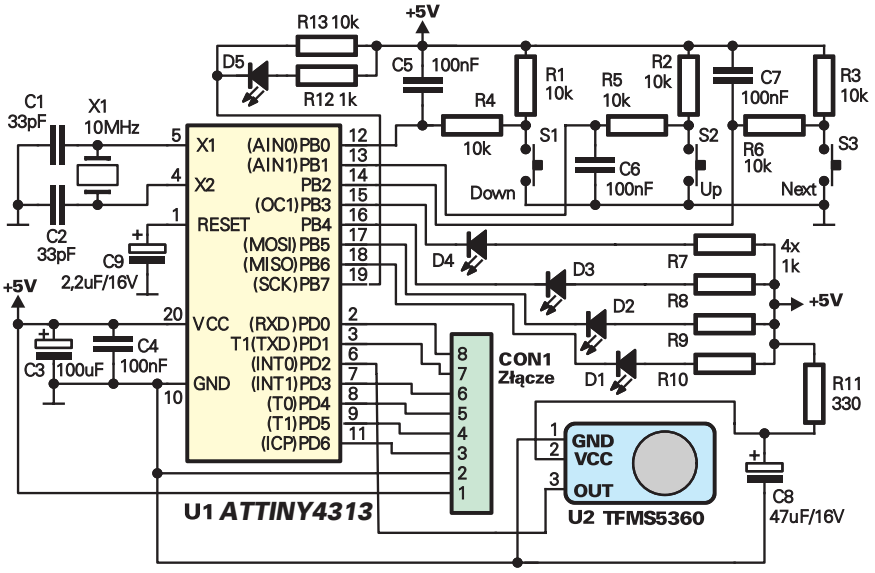
Całe urządzenie zostało podzielone na dwie części – sterującą i zasilająco-wykonawczą. **Rysunek 2** przedstawia schemat ideowy sterownika. Sercem układu jest procesor typu ATtiny4313 pracujący z kwarcem o częstotliwości 10MHz. Przyciski S1..S3 tworzą klawiaturę, dzięki której możliwe jest sterowanie układem bez pilota. Na temat funkcji S1..S3, a także o D1..D5 napisano w śródtytule *Obsługa*. Rezystor R11 wraz z kondensatorem C8 tworzą filtr zasilający odbiornik podczerwieni TFMS5360. Kondensator C9 dodano po testach pierwszego modelu - nie jest on potrzebny do prawidłowego resetowania układu po włączeniu napięcia zasilania, gdyż procesory AVR mają wbudowane układy zapewniające prawidłowy start. Jest on natomiast potrzebny do blokowania końcówki RESET, gdyż okazało się, że układ potrafił zresetować się samoczynnie przy włączaniu/wyłączaniu światła. Kondensatorek ten nie dopuszcza do takiej sytuacji.

Złącze CON1 służy do połączenia sterownika z płytką wykonawczą. Elementy C3, C4 blokują napięcie zasilania.

Przejdźmy teraz do **rysunku 3**, na którym widoczny jest schemat ideowy części zasilająco-wykonawczej. Elementy TR1, BR1, D1, C1...C4 oraz U1 tworzą typowy zasilacz stabilizowany o napięciu 5V. Wyjaśnienia wymaga tylko zastosowanie diody D1. Otóż dzięki niej w momencie zbliżania się napięcia sieci do zera tranzystor T1 zostaje zatkany. Czas zatkania tego tranzystora wynosił w modelu około 600µs. Dzięki temu procesor „wie", kiedy napięcie w sieci przechodzi przez zero. Złącze CON1 służy do podłączenia napięcia zasilania z sieci.

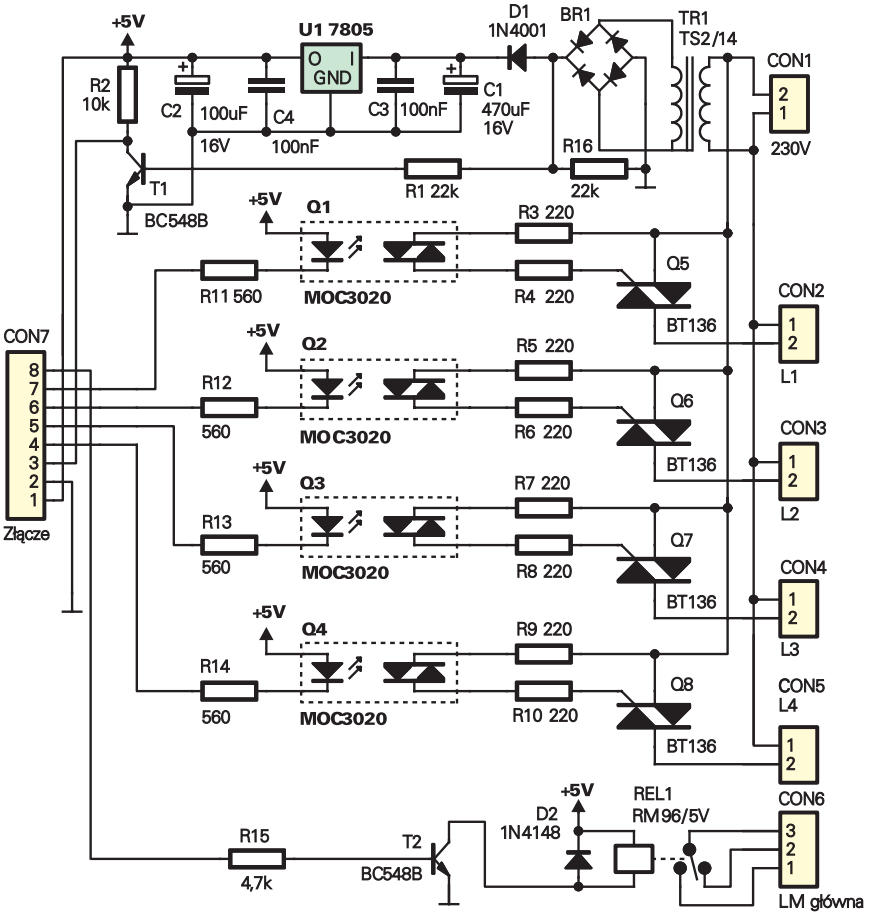
Optotriaki Q1..Q4 oraz triaki Q5..Q8 wraz z rezystorami R3..R14 tworzą część wykonawczą zasilającą lampy. Zastosowane optotriaki nie posiadają oczywiście obwodów włączania przy zerze w sieci. Tranzystor T2 steruje przekaźnikiem do załączania głównego oświetlenia. W tym obwodzie zdecydowałem się na użycie przekaźnika z tego względu, iż umożliwia on sterowanie oświetlaniem także za pomocą klasycznego przełącznika. Nie musimy rezygnować więc z włącznika światła głównego, do którego wszyscy są przyzwyczaili. **Rysunek 4** pokazuje, jak dołączyć typowy wyłącznik do urządzenia, podobne połączenia stosowane są często np. na klatkach schodowych.

Złącze CON7 służy do połączenia tej części ze sterownikiem.

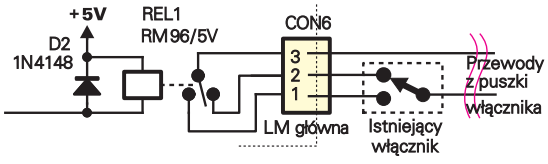


Rys. 2 Schemat ideowy sterownika

Rys. 3 Schemat ideowy układu wykonawczego



Rys. 4 Sposób dołączenia włącznika światła do układu



Program sterujący

Jak już pisałem we wstępie, program został napisany w Bascomie AVR. Składa się on z wielu podprogramów, co na początku może wydać się dziwne. Jednak taki podział programu pozwolił na zmieszczenie go w pamięci '4313. Kod źródłowy wzbogacony jest komentarzami, dzięki czemu nie powinno być większych problemów z jego zrozumieniem.

Podstawą działania układu są bardzo częste przerwania od Timera1. Procesor skacze

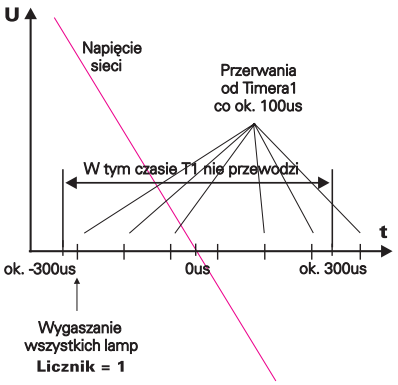
do podprogramu obsługi tych przerwań co około 100µs, co daje mniej więcej 100 kroków regulacji jasności. Podprogram ten przedstawia **Listing 1**. Zmienna *Licznik* odlicza, ile czasu upłynęło od przejścia napięcia sieci przez zero. Natomiast czterobajtowa tablica *Lmram* przechowuje informację o aktualnej jasności każdej lampy. Dodatkowo w podprogramie tym sprawdzany jest stan końcówki PD.6, do której jest dołączony tranzystor T1. Jeśli stan tej końcówki wynosi „1", to *Licznik* zostaje wyzerowana, a lampy wygaszone. Pomocą w zrozumieniu tego fragmentu jest **rysunek 5**, na którym widoczny

Listing 1

```
Inttim1:
  Portd.0 = Onofflmgl.0                          'zświecenie/gaszenie lampy głównej
  Timer1 = 64566                                 'timer1 liczy 970 cykli zegara, czyli
                                                 'przerwanie co około 97us
  If Licznik >= 50 Then
    $asm
      sbic pind, 6                               'sprawdzaj stan końcówki PIND6, jeśli jest
                                                 'równa 1, to
    $end Asm
      Gosub Zero                                 'skocz do podprogramu Zero
  End If
  Incr Licznik
  If Onoffall = 255 Then             'jeśli zezwolenie_globalne_na_włączanie_lamp = 1, to
    If Licznik > Lmram(1) Then Reset Portd.5    '}
    If Licznik > Lmram(2) Then Reset Portd.4    '  } zapalaj odpowiednio lampy do wartości
                                                 'tablicy Lmram
    If Licznik > Lmram(3) Then Reset Portd.3    '}
    If Licznik > Lmram(4) Then Reset Portd.1    '}
  End If
Return
(......)
Zero:
  Licznik = 1                                    'przy każdym przechodzeniu napięcia sieci
                                                 'przez zero zeruj licznik i
  Portd = Portd Or &B1111110                     'wygaś wszystkie 4 lampy
  Incr Timefader
Return
```

Listing 2

```
If Address = Eaddress Then                'jeśli adres pilota dobry
    For Nextlm = 1 To 16                  'szesnastokrotnie
      If Command = Epamiec(nextlm) Then   'sprawdzanie komend przycisków pilota
       Select Case Nextlm
         Case 1 : Gosub Incrktora         'Incr Która
         Case 2 : Gosub Setupbit          'przycisk Up
         Case 3 : Gosub Setdownbit        'przycisk Down
         Case 4 : If Kod = 0 Then         'po to, by jednokrotnie zmienić stan onofflmgl
                                          'przy jednokrotnym naciśnięciu przycisku pilota
                    Onofflmgl = Not Onofflmgl
                    Eonofflmgl = Onofflmgl
                    Gosub Setkod
                  End If
         Case 5 : If Kod = 0 Then         'po to by jednokrotnie zmienić stan onoffall przy
                                          'jednokrotnym naciśnięciu przycisku pilota
                    Onoffall = Not Onoffall
                    Gosub Setkod
                  End If
         Case 6 : If Kod = 0 Then         'po to, by jednokrotnie zmienić stan DDRB.7
                                          'przy jednokrotnym naciśnięciu przycisku pilota
                    Ddrb.7 = Not Ddrb.7       'przełączanie PORTD.7 jako we. lub
                                          'wy. co daje włączanie i wyłączanie Timer Fadera
                    Gosub Setkod
                  End If
         Case Is >= 7 :
                              Incr X        'zwiększanie zmiennej X
                              Adr = Nextlm  '}obliczanie adresu dla pamięci eeprom]
                                                    Shift Adr , Left , 2
                              If X = 10 Then     'po ok 1s przyciskania klawisza
                                For Digit = 1 To 4
                                  Writeeeprom Lmram(digit) , Adr    'zapis do eeprom
                                      'wartości jasności lamp danego profilu
                                  Gosub Waitms50 'czas dla EEPROMa na zapis
                                  Incr Adr
                                Next Digit
                                Gosub Mrugnij
                                Pomocbit3 = 0
                              Else
                                Pomocbit3 = 255  'blokowanie odczytu z EEPROM jeśli
                                                 'przycisk jest trzymany
                              End If
       End Select
      End If
    Next Nextlm
(........)
End If
```



Rys. 5 Wyjaśnienie działania układu

jest w dużym powiększeniu (w stosunku do Rys.1) moment przechodzenia napięcia sieci przez zero. Kolejne zapalenie lamp nastąpi po czasie wyznaczonym przez *Lmram*. Bajt *Onoffall* decyduje o tym, czy lampy mają być zapalane, czy też nie, natomiast bajt *Onofflmgl* decyduje o stanie przekaźnika sterującego oświetleniem głównym. Ten ostatni po każdej zmianie jest zapamiętywany w pamięci EEPROM. Chodzi przy tym o to, że lampa dołączona do przekaźnika może być zapalana i gaszona za jego pomocą lub też za pomocą klasycznego włącznika umieszczonego w puszce. Jeśli np. w nocy lub podczas naszej nieobecności wyłączono by na chwilę napięcie w sieci, to istnieje 50% szans, że po powrocie tego napięcia światło byłoby niepotrzebnie zapalone. Wpisanie do pamięci EEPROM ostatniego stanu przekaźnika zapobiega takiej sytuacji. Zmienne takie jak *Onoffall* czy *Onofflmgl* przyjmują tak naprawdę tylko wartości bitowe, w ich przypadku 0 lub 255. Po co więc takie marnowanie pamięci RAM? Otóż wynika to z prostego faktu: operacje na liczbach 8-bitowych są dla naszego µP znacznie prostsze, a co najważniejsze, wymagają znacznie mniej pamięci FLASH. Gdyby wszystkie zmienne podobne do *Onoffall* i *Onofflmgl* zmienić na bity, zyskalibyśmy trochę wolnej pamięci RAM (co i tak nic nie zmienia), ale program nijak nie zmieściłby się w naszym '2313.

Zwróćmy jeszcze uwagę na **Listing 2**, na którym przedstawiono fragment kodu odpowiedzialny za sprawdzanie, czy adres i odebrana komenda z pilota to jedna z wcześniej zapisanych. Jeśli tak, to procesor podejmuje stosowne kroki. Przykładowo, jeśli zostanie wciśnięty przycisk o numerze 7...16, czyli jeden z przycisków profili, program obliczy adres dla pamięci EEPROM, pod którym ma zostać zapisany dany profil. Zapis do EEPROM nie nastąpi jednak od razu, za pewne opóźnienie odpowie-

dzialna jest zmienna *X*. Wpisanie nowych danych profilu następuje dopiero po około 1s trzymania klawisza na pilocie. Jeśli jednak przycisk nie będzie trzymany na tyle długo, by *X* osiągnął wartość 10, do pamięci EEPROM nic nowego nie zostanie zapisane. Wywoła to odwrotny skutek, tzn. tym razem dane zostaną odczytane z EEPROM (w dalszej części kodu) i zapisane w tablicy *Lmram*. Podsumujmy: jeśli przycisk danego profilu został naciśnięty raz, dane zostaną odczytane z pamięci EEPROM i wpisane do tablicy *Lmaram*. Jeśli jednak trzymamy przycisk dłużej niż 1s, to nowe dane jasności poszczególnych lamp zostaną zapisane w danym profilu.

Pozostała część programu zajmuje się odczytywaniem stanu klawiatury i sczytywaniem danych z pilota, a także odpowiednim zapalaniem diod D1...D4.

Montaż i uruchomienie

Montaż układu przeprowadzamy na płytkach drukowanych, widocznych na **rysunkach 6** i **7**. W przypadku płytki sterownika należy w pierwszej kolejności wlutować zworki w miejsce rezystora R4, a także jedną zworkę oznaczoną Z. Resztę elementów na obydwu płytkach montujemy w typowy sposób, dla własnej wygody zaczynając od elementów najmniejszych. Układ modelowy umieszczono w obudowie Z-46, dla której została przygotowana płyta czołowa widoczna na **rysunku 8**. Na papier z nadrukiem warto nakleić folię samoprzylepną, co zabezpieczy go przed zabrudzeniem. Przypominam też o konieczności wycięcia otworku dla układu U2.

Na koniec, jeśli mamy już odpowiednio przygotowaną obudowę, łączymy przewodami obydwie płytki. Można to wykonać za pomocą specjalnej złączki lub, tak jak w modelu, lutując przewody do goldpinów. Zwróćmy uwagę na odpowiednią kolejność przewodów, która została zaznaczona na płytkach. Następnie łączymy przewodami elementy D1...D4, S1...S3, a także układ U2, który został przyklejony klejem na ciepło do obudowy. Do stabilizatora 7805 należy przykręcić kawałeczek blaszki, by przy włączonych lampach i przekaźniku zbyt mocno się nie nagrzewał.

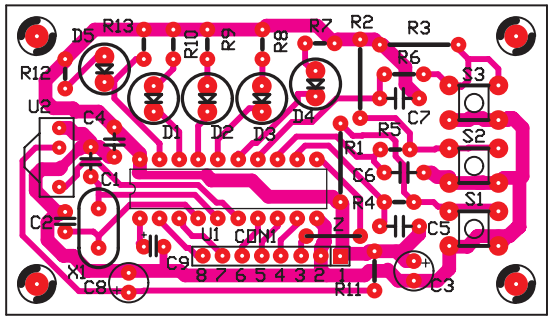
Po **dokładnym sprawdzeniu** poprawności montażu można przystąpić do włączenia układu do sieci, **należy przy tym zachować szczególną ostrożność**. Przed włożeniem do podstawki mikroprocesora warto sprawdzić, czy między jego 10 a 20 pinem napięcie wynosi 5V. Jeśli tak jest, wyłączamy zasilanie i wkładamy µP w podstawkę U1. Jeśli wszystko zamontowaliśmy poprawnie, to po ponownym włączeniu układu do sieci powinna zaświecić się dioda D1. Naciskanie przycisku S3 (Next) powodować będzie zapalanie kolejnych LED-ów.

Moc żarówek sterowanych triakami może dochodzić do 150W bez konieczności stosowania radiatora. Jeśli ktoś jednak będzie chciał niewielki zamontować, podkreślam konieczność zastosowania podkładek izolujących pod wszystkie triaki.

Obsługa

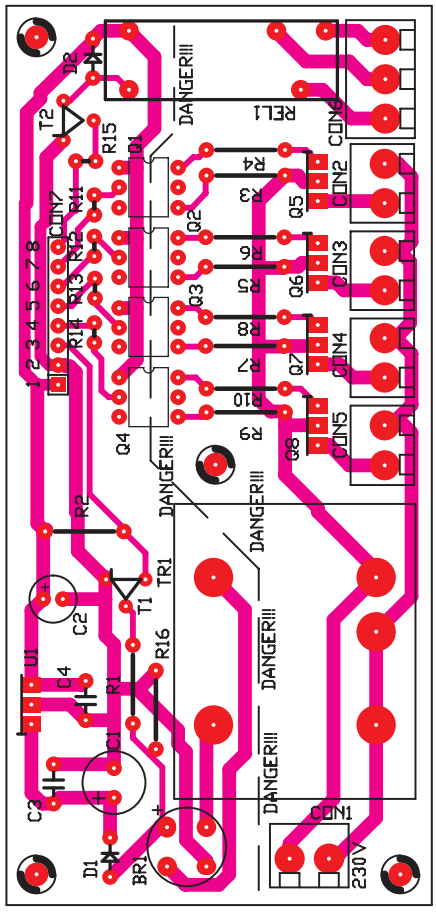
Na początku omówię funkcję diod LED, a także przycisków S1...S3. Każda dioda LED odpowiada jednej lampie, tzn. jeśli świeci się D1, to za pomocą przycisków Up i Down możemy zmieniać jasność lampy pierwszej, jeśli D2 – drugiej itd. Jeśli dojdziemy jednak do zaświecenia diody D4, następne wciśnięcie Next spowoduje zaświecenie wszystkich LED-ów. Jest to tryb, w którym możemy sterować jasnością wszystkich lamp jednocześnie, tzn. bieżąca jasność każdej lampy jest zmieniana proporcjonalnie do czasu naciskania przycisków Up i Down. Kolejne wciśnięcie Next da w efekcie wygaszenie wszystkich LED-ów. Tryb ten służy do zaświecania i gaszenia sterowanych lamp, a także tej podłączonej do przekaźnika. Zaświecania i gaszenia dokonujemy przyciskami Up i Down. Kolejne wciśnięcie Next spowoduje zaświecenie diody D1, czyli powrót do stanu początkowego. Układ może być pozostawiony oczywiście w dowolnym trybie.

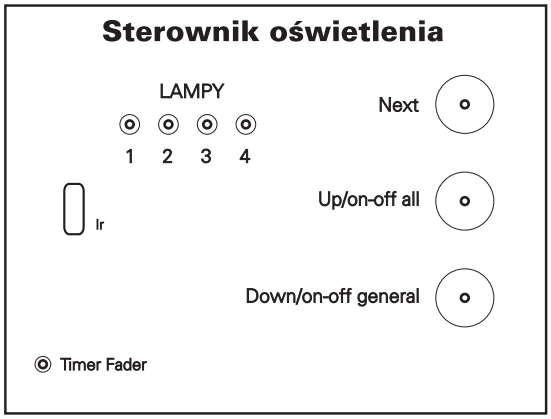
Jeśli dysponujemy pilotem, możemy przy-



Rys. 6 Schemat montażowy sterownika

Rys. 7 Schemat montażowy układu wykonawczego





Rys. 8 Płyta czołowa (skala 50%)

stąpić do nauczenia układu reagowania na odpowiednie komendy. W tym celu należy włączyć zasilanie układu, trzymając wciśnięty przycisk S1 (Down), tym razem podobnie jak poprzednio zaświeci się dioda D1 i układ będzie czekał na komendy pilota. Wciskamy następnie wybrane przyciski na pilocie w następującej kolejności: Next; Up; Down; włącz./wył. - sterowane lampy; włącz./wył. - lampa główna; Timer Fader; dziesięć przycisków profili. Zapamiętywanie przez układ kolejnych komend będzie sygnalizowane mignięciem diody D1. Po tym procesie adres i komendy naszego pilota zostaną zapamiętane w pamięci EEPROM. Nie musimy zapamiętywać wszystkich 16 przycisków, jeśli wystarczą nam np. 4 profile, proces ten kończymy po wciśnięciu 10 przycisku. Następnie należy ponownie wyłączyć na chwilkę zasilanie i ponownie je włączyć, tym razem normalnie, czyli już bez trzymania przycisku Down. Teraz możemy sterować układem także za pomocą pilota. Rola przycisków Next, Up i Down jest prawie identyczna jak dla klawiatury, z tą różnicą, że do zaświecania i gaszenia lamp mamy oddzielne przyciski na pilocie. Funkcję Timer Fader możemy włączyć/wyłączyć w dowolnym momencie. Funkcja ta będzie działała na tę lampę, której dioda będzie zaświecona. Jeśli chcemy, by wszystkie lampy były powoli wygaszane, musimy ustawić tryb, w którym świecą się wszystkie diody. Ustawione profile zapamiętujemy w dowolnym momencie poprzez dłuższe przytrzymanie jednego z przycisków profili na pilocie. Zapamiętanie będzie zasygnalizowane mrugnięciem diody D1. Zapamiętane profile wywołujemy poprzez zwyczajne naciskanie odpowiednich przycisków. Funkcja profili jest bardzo przydatna, w praktyce okazuje się bowiem, że zmiany oświetlenia dokonywane są najczęściej właśnie poprzez wywoływanie odpowiednich profili. Jest to bardzo wygodne.

Możliwości zmian i rozbudowy

Jeżeli ktoś nie ma potrzeby sterowania aż czterema lampami, może nie montować wszystkich triaków i optotriaków. W takim wypadku można zmienić fragment kodu programu tak, by „omijał" niewykorzystane lampy. Zmiana taka nie jest konieczna, jednak da większą wygodę obsługi. Podpowiem więc, że wystarczy dokonać drobnych zmian w podprogramie *Selectora*, a także skrócić cykl licznika *Ktora.*

Dzięki nieco dziwnemu sterowaniu włączaniem i wyłączaniem funkcji Timer Fader, możemy w prosty sposób zmienić układ w łagodny budzik. Normalnie, gdy funkcja automatycznego ściemnienia jest nieaktywna, PB.7 jest wejściem „pływającym". Poprzez rezystor R13 zostaje podciągnięte do plusa zasilania. Jednak wciśnięcie odpowiedniego przycisku na pilocie zmienia bit DDRB.7, a tym samym ustawia PB.7 jako wyjście. Tym razem na PB.7 pojawi się masa. Program sprawdza w podprogramie pętli głównej stan bitu PinB.7 i jeśli jest w nim „0", uaktywnia Timer Fader. Z tego wynika więc, że włączenia funkcji Timer Fader można dokonać również poprzez zwarcie końcówki PB.7 do masy. Jeśli ktoś nie posiada pilota, może uruchamiać tę funkcje poprzez np. dodatkowy przełącznik. Idąc dalej, możemy zmienić jedną linijkę w podprogramie *Timefadsub* (*Gosub Setdownbit* na *Gosub Setupbit*), by uzyskać efekt odwrotny tzn. powolnego rozjaśniania światła. Podłączając do PB.7 poprzez tranzystorek NPN z otwartym kolektorem wyjście budzika, możemy stać się posiadaczami układu, który będzie nas dodatkowo budził łagodnie rozjaśnianym światłem. Dobrze by było tak zgrać układy, by dźwięk budzenia odzywał się wtedy, gdy lampy będą już lekko rozjaśnione.

Uwaga! Podczas użytkowania przyrządu w jego obwodach mogą wystąpić napięcia groźne dla życia i zdrowia. Osoby niepełnoletnie mogą wykonać i przetestować przyrząd wyłącznie pod opieką wykwalifikowanych opiekunów.

Kamil Kozłowski
Rafał Mankiewicz

Wykaz elementów

Płytka sterownika

Rezystory
R1,R5,R6 ..... 10kΩ
R2, R3, R4, R13 ..... 10k Ω
R7-R10,R12 ..... 1kΩ
R11 ..... 330Ω

Kondensatory
C1,C2 ..... 33pF
C3 ..... 100µF/16V
C4-C7 ..... 100nF
C8 ..... 47µF/16V
C9 ..... 2,2µF/16V

Półprzewodniki
D1-D4 ..... diody LED 5mm
D5 ..... dioda LED 3mm
U1 ..... ATTINY4313 (zaprogramowany)
U2 ..... TFMS5360

Inne
X1 ..... rezonator kwarcowy 10MHz
CON1 ..... listwa goldpin
S1-S3 .. przełączniki chwilowe
podstawka DIL20

Płytka układu wykonawczego

Rezystory
R1,R16 ..... 22kΩ
R2 ..... 10kΩ
R3-R10 ..... 220Ω
R11-R14 ..... 560Ω
R15 ..... 4,7kΩ

Kondensatory
C1 ..... 470µF/16V
C2 ..... 100µF/16V
C3,C4 ..... 100nF/ceramiki

Półprzewodniki
D1 ..... 1N4001
D2 ..... 1N4148
BR1 ..... mostek 1A
T1,T2 ..... BC548
U1 7805
Q1-Q4 ..... MOC3020
Q5-Q8 ..... BT136

Inne
CON7 ..... listwa goldpin
CON1-CON5 ..... złącze śrubowe ARK2
CON6 złącze śrubowe ARK3
REL1 ..... przekaźnik RM96P/5V
TR1 ..... TS2/14

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2749.