

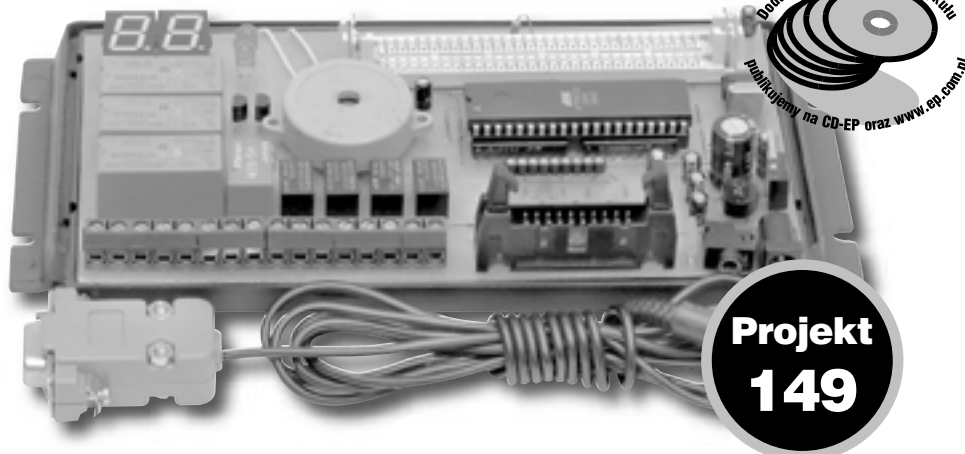
Dział „Projekty Czytelników” zawiera opisy projektów nadesłanych do redakcji EP przez Czytelników. Redakcja nie bierze odpowiedzialności za nieprawidłowe działanie opisywanych układów, gdyż nie testujemy ich laboratoryjnie, chociaż sprawdzamy poprawność konstrukcji.

Prosimy o nadsyłanie własnych projektów z modelami (do zwrotu). Do artykułu należy dołączyć podpisane oświadczenie, że artykuł jest własnym opracowaniem autora i nie był dotychczas nigdzie publikowany. Honorarium za publikację w tym dziale wynosi 250,- zł (brutto) za 1 stronę w EP. Przesyłanych tekstów nie zwracamy. Redakcja zastrzega sobie prawo do dokonywania skrótów.

Uniwersalny sterownik 16-kanalowy sterowany z PC, część 1

Do naszych domów na dobre wkradła się elektronika. Nie chodzi już o tak oczywiste wyposażenie, jak odbiornik telewizyjny czy radio; przybywa nam wszelkiego rodzaju urządzeń mniej lub bardziej przydatnych, są i kłopotliwe czy wymagające nadzoru lub sterowania. Grupa ta obejmuje takie „gadżety”, jak automatyczne bramy, elektrycznie sterowane rolety, sterowane światliki dachowe, oświetlenie ogrodowe, klimatyzatory, systemy wentylacji, systemy ogrzewania i wiele, wiele innych. Często urządzenia te działają samodzielnie, tzn. posiadają własne obwody nadzoru czy sterowania. Obsługa sprowadza się właściwie do ręcznego ich włączenia czy zmieniania nastaw. Niniejszy sterownik został zbudowany właśnie z myślą o ich zdalnym sterowaniu, bez odchodzenia od komputera.

Rekomendacje:
uniwersalne urządzenie sterujące o dużej liczbie kanałów, łatwe w montażu i obsłudze, przeznaczone do zastosowań ogólnych.



Urządzenie obsługuje 16 niezależnych kanałów, z czego 8 zrealizowałem jako wyjścia przekaźnikowe (styki robocze przekaźników typu „Normally Open”), a pozostałe 8 kanałów jako wyjścia bezpośrednie, wystawiające poprzez drivery poziom +12 V. Do prostej sygnalizacji stanu kolejnych kanałów zastosowałem podwójny wyświetlacz oraz diody LED. Urządzenie zasilane jest właściwie z dowolnego zasilacza niestabilizowanego o napięciu z przedziału 10...16 V/300 mA. Napięcie to ma wpływ na poziom napięcia na wyjściowych kanałach 9...16. Sterownik jest dodatkowo wyposażony w buzzer sygnalizujący zmiany stanu poszczególnych kanałów oraz w pamięć EEPROM zastosowaną w celu zapamiętywania ostatnich nastaw (na wypadek przerw w dostarczaniu energii elektrycznej – w moim wykonaniu jest to opcjonalne). W urządzeniu przewidziałem także możliwość zainstalowania dodatkowych modułów, jak choćby odbiornika popularnego pasma 433 MHz – i w takim przypadku uzyskuje się bezprzewodowy system sterowania. Do sterowania urządzeniem wykorzystuję wolny port COM w komputerze, prosty w obsłudze i – jak do dziś – praktycznie wszechobecny. Komunikacja ze sterownikiem jest banalna, polega mianowicie na wysłaniu do urządzenia kompletnych znaków ASCII. Sterownik oczekuje na

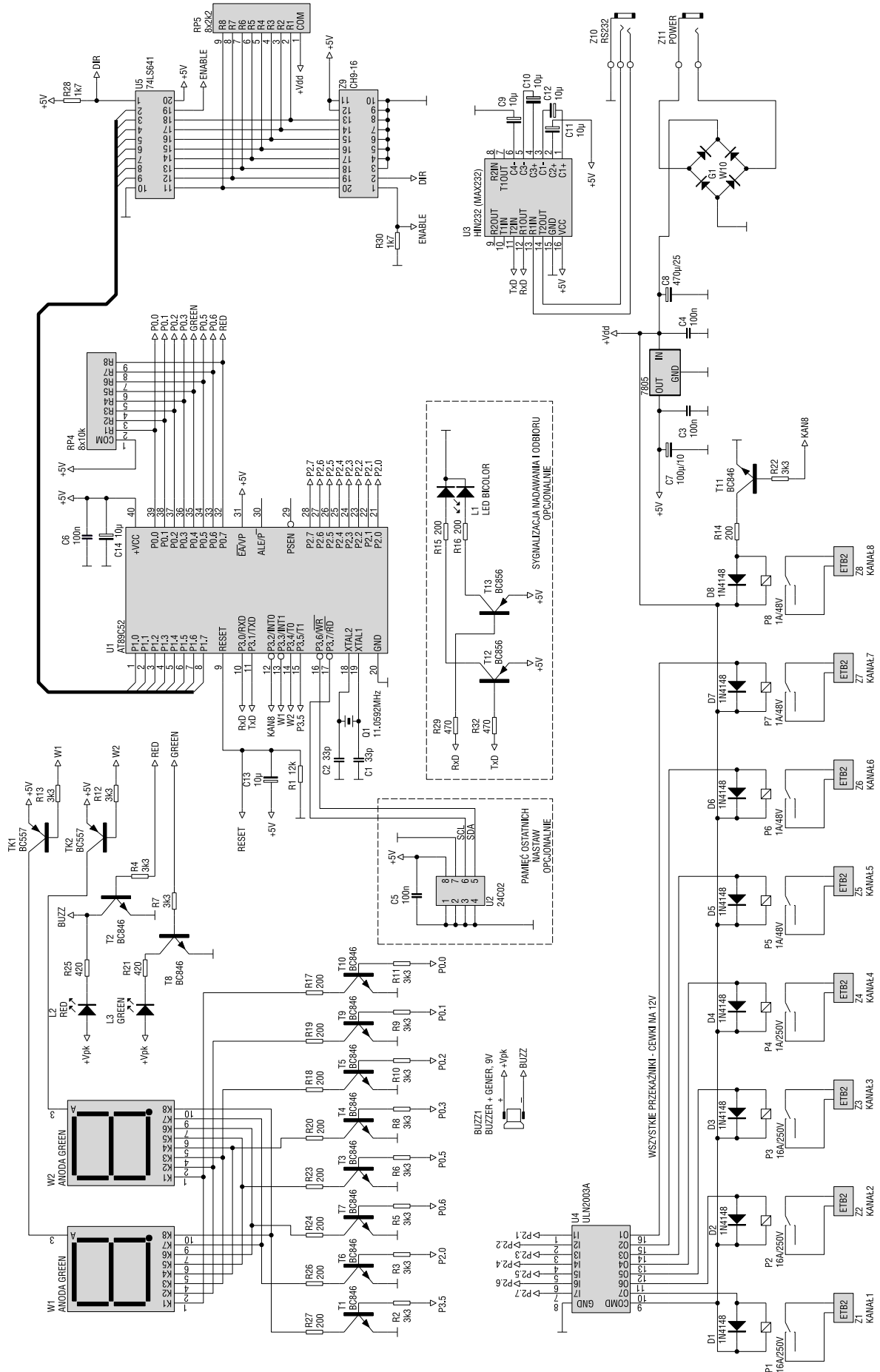
znak i interpretuje go. Na każdy kanał przypadają dwa znaki odpowiedzialne za włączenie i wyłączenie danego kanału oraz dodatkowo znak, który blokuje dalsze przyjmowanie sygnałów sterujących (tzw. „LOCK”). Tym samym znakiem wyprowadzamy także sterownik ze stanu „LOCK”.

Opis układu

Schemat sterownika przedstawiono na rys. 1. Serduszkim układu jest popularny i tani mikrokontroler ATMEL AT89C52. Pracuje w typowej konfiguracji sugerowanej przez producenta. Rezonator kwarcowy ma wartość 11,05920 MHz, co jest istotne – to musi być taka wartość; inaczej nasz sterownik nie będzie rozpoznawał znaków nadchodzących z PC, ponieważ czasy ramki nie będą się zgadzały! Nasz „atmelek” ma sporo portów, więc zrezygnowałem z wyszukanych rozwiązań zwielokrotniających liczbę portów. Układ jest dzięki temu prosty i zrozumiały. Z mikrokontrolerem współpracuje konwerter stanów RS232 na standard TTL. Jest nim tani odpowiednik znanego układu MAX232 (HIN232CP). To on dokonuje konwersji i tym samym łączy środowisko sterownika z „pecetem”. Dodałem także dwukolorowego LED-a po stronie poziomów TTL, by w oczywisty sposób

PODSTAWOWE PARAMETRY

- Płytki o wymiarach 163 x 87 mm
- Zasilanie: 10...16 V/300 mA (niestabilizowane)
- Liczba kanałów przekaźnikowych: 8
- Liczba kanałów napięciowych: 8
- Napięcie wyjściowe kanałów napięciowych: +12 V
- Sygnalizacja stanu: wyświetlacz LED + diody LED
- Dźwiękowa sygnalizacja zmiany stanu kanałów
- Pamięć nastaw EEPROM
- Możliwość współpracy z modulem radiowym
- Komunikacja z komputerem przez RS232
- Możliwość blokady poszczególnych kanałów (funkcja „LOCK”)



Rys. 1. Schemat elektryczny sterownika

Tab. 1. Przyporządkowania znaków do funkcji

Spacja	Zablokuj/odblokuj sterownik		
0 (zero)	Wyłącz wszystkie kanały		
1	Włącz kanał mocy 1	a	Ustaw kanał 9 (wystaw +12 V)
2	Włącz kanał mocy 2	s	Ustaw kanał 10
3	Włącz kanał mocy 3	d	Ustaw kanał 11
4	Włącz kanał mocy 4	f	Ustaw kanał 12
5	Włącz kanał mocy 5	g	Ustaw kanał 13
6	Włącz kanał mocy 6	h	Ustaw kanał 14
7	Włącz kanał mocy 7	j	Ustaw kanał 15
8	Włącz kanał mocy 8	k	Ustaw kanał 16
q	Wyłącz kanał mocy 1	z	Zeruj kanał 9 (wystaw 0 V)
w	Wyłącz kanał mocy 2	x	Zeruj kanał 10
e	Wyłącz kanał mocy 3	c	Zeruj kanał 11
r	Wyłącz kanał mocy 4	v	Zeruj kanał 12
t	Wyłącz kanał mocy 5	b	Zeruj kanał 13
y	Wyłącz kanał mocy 6	n	Zeruj kanał 14
u	Wyłącz kanał mocy 7	m	Zeruj kanał 15
i	Wyłącz kanał mocy 8	,	Zeruj kanał 16
Uwaga! Sterownik oczekuje małych liter! (Wielkie litery mają inne kody ASCII)			

wiedzieć, czy „pecet” wysyła do sterownika znaki i czy sterownik odpowiada na nie. Także potraktowałem to jako opcję w prezentowanym projekcie. LED po prostu miga w czasie transmisji znaków. Ponieważ transmisja wymaga właściwie 3 przewodów, zastosowałem zwykle gniazdo „JACK” stereo: masa, RxD i TxD. Oszczędziłem w ten sposób miejsce na płytce. Wprost z portu P0 jest sterowany bufor mocy ULN2003A (właściwie są to tranzystory Darlingtona we wspólnej obudowie) i to on załącza kolejne przekaźniki. Cewki przekaźników są blokowane diodami impulsowymi 1N4148, które wygaszają zaindukowane napięcie niedopuszczając tym samym do powstawania groźnych przepięć. Do wygodnego połączenia przekaźników z urządzeniami sterowanymi służy rząd bloczków (terminal block). Pierwsze 8 kanałów zrealizowałem w podobny sposób, zmieniają się tylko przekaźniki – są 3 na dość spory prąd, jeden mniejszy i cztery miniaturowe na niewielkie wartości prądów. Na pozostałych 8 kanałach wystawiane jest napięcie lub wymuszana masa. W momencie opracowywania układu stwierdziłem, że w wielu przypadkach może się to przydać np. do wysterowania innych oddalonych przekaźników czy „zadawania” napięcia sterującego. Zatem port P1 jest połączony z nadajnikiem/odbiornikiem linii 74LS641, który wystawia napięcie na złącze wyjściowe. Pamiętajmy jednak, że wartość prądu na tych kanałach nie jest duża,

właściwie jest to dobre do sterowania np. transoptorami. W założeniu chciałem utrzymać kontekst sterownika możliwie uniwersalnego, więc zdecydowałem się na nadajnik/odbiornik z wyjściami typu otwarty kolektor, co w połączeniu z wyprowadzonym napięciem +5 V na złącze kanałów 9...16 daje możliwość załączania większych obciążeń „na masie”. W tym układzie możemy otrzymywać natężenie prądu do około 20 mA na kanał, gdyż tyle możemy wycisnąć z wyjścia typu otwarty kolektor tej kostki. Kilka słów o złączach wewnątrz sterownika: zakładałem dobudowanie odbiornika 433 MHz i ewentualnie modułu automatycznych sekwencji lub watchdoga, wmontowałem więc dosyć spore złącze z... SIM-30 starych płyt głównych używanych tam jako miejsce instalowania ówczesnych kostek pamięci. Oczywiście nic nie stoi na przeszkodzie, aby zastosować inne złącze (albo je pominąć). Układ ma charakter prototypowy, zaprojektowałem obwód jako jednostronny (nie udało się uniknąć kilku mostków) oraz zastosowałem elementy w technologii SMT. W proponowanym rozwiązaniu znajdziemy także pozornie zbędne drabinki rezystancyjne („R-Pack-i”), które w standardowym wykonaniu są rzeczywiście niekoniczne, gdyż mikrokontroler ma wbudowane już rezystory podciągające wewnątrz swojej struktury. Tylko port P0 wymaga zewnętrznej drabinki, gdyż jest on skonstruowany jako port wejścia/wyjścia o wy-

sokiej impedancji. Mniejsze scalaki są lutowane od strony ścieżek na podgiętych nogach, co uznałem jako ciekawe i pozwalające ograniczyć liczbę otworów w płytce drukowanej.

Program mikrokontrolera

Oprogramowanie sterownika zostało napisane w assemblerze, zapewniając tym samym oszczędny i wydajny kod. Po włączeniu urządzenia sterownik wykonuje proste procedury testowe i odczytuje zawartość EEPROM (jeśli pamięci nie zamontujemy, to ten krok jest pomijany). Następnie zaczyna „meldunek” stanu swoich portów, wyświetlając numer kanału z równoczesnym sygnalizowaniem jego stanu. Stan aktywny (włączony) dodatkowo sygnalizowany jest krótkim piskiem z buzzera. Diody LED: zielona – kanał wyłączony, czerwona – kanał włączony. Po zakończeniu meldunku sterownik przechodzi do stanu oczekiwania na nadejście znaku. Sygnalizowane jest to okresowym migotaniem środkowych segmentów na obu wyświetlaczach. Kiedy prześlemy kod do sterownika, wyświetla on numer kanału, który został włączony/wyłączony oraz potwierdzenie wykonania dyspozycji „EC”. Oczywiście, przy rozkazie „włącz kanał” usłyszymy także pisk. W czasie kiedy sterownik się nudzi, raportuje jeszcze dwukrotnie (w odstępach około minuty) stan portów, przy czym po drugim raportowaniu sam się blokuje i sygnalizuje ten stan poprzez cykliczne wyświetlanie „Lo.”. W przypadku błędów odczytu znaku nadchodzącego, sterownik kwituje to symbolem „Er.” Może się tak stać z kilku przyczyn: złe ustawienie parametrów transmisji, nieodpowiedni kod dyspozycji (nierozpoznawalny przez sterownik) lub próba ponownego załączenia już aktywnego kanału (lub 2 razy z rzędu wyłączenie tego samego kanału). Do kodów dyspozycji dodałem także awaryjny kod „0” – sterownik wyłącza natychmiast wszystkie kanały symbolem: „A.O.” (All Off). W stanie zablokowania sterownik nie reaguje na wysyłane kody sterujące, chyba że jest to kod odblokowania (kod spacji w ASCII). I właściwie to już wszystko.

Adrian Kuglin
adrian1978@tlen.pl