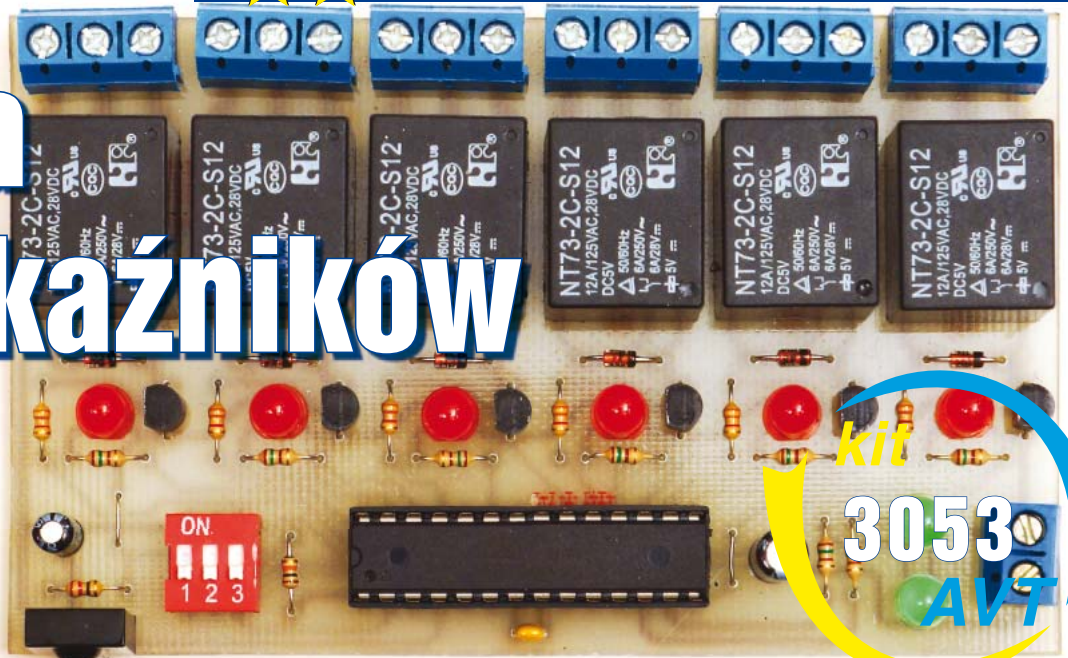




Karta przełączników RC5



kit 3053

AVR

Do czego to służy?

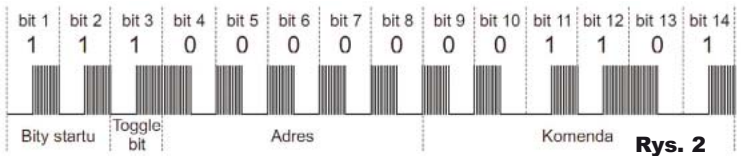
Prezentowany układ służy do bezprzewodowego sterowania za pośrednictwem pilota RC5 dowolnymi sześcioma urządzeniami, podłączonymi do przełączników. Urządzenie pozwala na skalibrowanie przycisków pilota przy pierwszym uruchomieniu lub w dowolnym momencie pracy. Wszystkie ustawienia są zapamiętywane w nieulotnej pamięci EEPROM. W dowolnym momencie pracy układu można zablokować możliwość sterowania wyjść mocy za pośrednictwem pilota. Ze względów bezpieczeństwa układ został także wyposażony w awaryjny wyłącznik wszystkich przełączników, bez konieczności stosowania pilota.

Jak to działa?

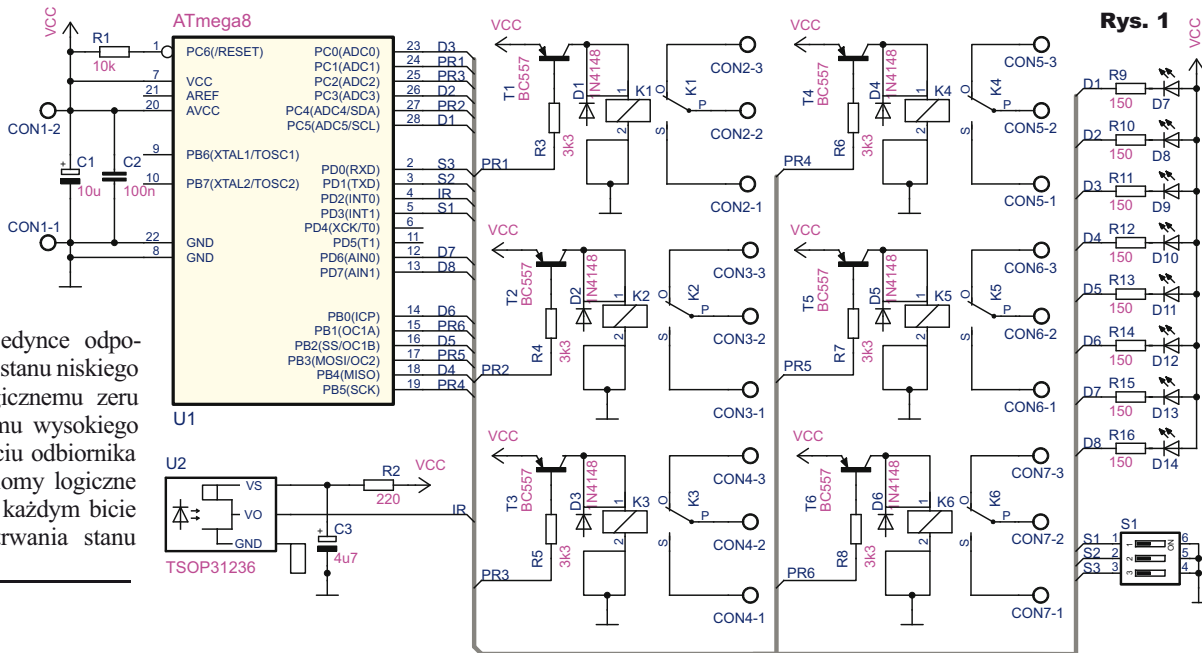
Schemat ideowy przedstawiono na rysunku 1. Układ scalony U1 (ATmega8) pracuje na wewnętrznym oscylatorze RC 1MHz w prostej, typowej aplikacji. Przełączniki są włączane kluczami tranzystorowymi, a diody D1–D6 zabezpieczają przed przepięciami. Scalony odbiornik podczerwieni TSOP31236 w z m a c n i a i demoduluje falę nośną o częstotliwości 36kHz. Standard RC5 opiera się na kodowaniu bifazowym (Manchester). Logicznej jedynce odpowiada przejście ze stanu niskiego na wysoki, a logicznemu zeru przejście z poziomu wysokiego na niski. Na wyjściu odbiornika podczerwieni poziomy logiczne są odwrócone. W każdym bicie (1,778ms) czas trwania stanu

wysokiego i niskiego wynosi po 889µs. Rysunek 2 przedstawia strukturę ramki standardu RC5. Pierwsze dwa bity to bity startu, które zawsze przyjmują wartość jeden. Trzeci bit to Togglebit, który zmienia wartość na przeciwną po każdym wciśnięciu danego przycisku. Kolejne pięć bitów zawiera adres, a ostatnie sześć komendę. Program sterujący mikrokontrolerem został napisany w środowisku BASCOM AVR. Na listingu 1 (dostępny w Elportalu) pokazano fundamentalną część programu wykonywaną w nieskończonej pętli. Na samym początku sprawdzany jest stan przełączników Dip Switch. W przypadku, gdy pierwszy przełącznik został zwarty, program przechodzi do podprogramu Ustawienia w celu

skalibrowania przycisków pilota, natomiast gdy zwarty został drugi przełącznik, program wyłącza wszystkie przełączniki i blokuje przerwanie INT0, aby zablokować możliwość sterowania urządzeniem za pośrednictwem pilota. Analogicznie wygląda reakcja na odczytanie zwarcia trzeciego przełącznika, z tym że program nie wyłącza przełączników, a jedynie blokuje możliwość sterowania pilotem. Zastosowane opóźnienia i powtórne sprawdzenie stanu przycisku wyklucza drgania zestyków. Kolejne instrukcje są wykonywane w przypadku, gdy użytkownik wciśnie dowolny klawisz, o czym informuje zmienna flagowa

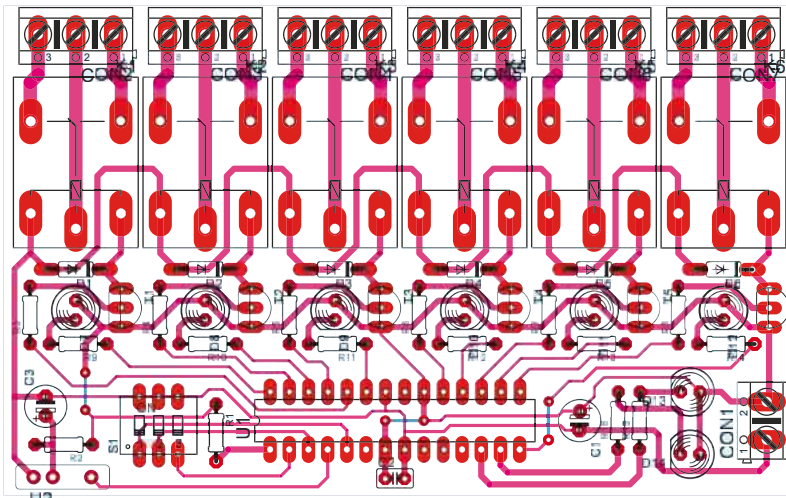


Rys. 2



Rys. 1

Odebr_kod. Wartość odebranej komendy jest porównywana z wartościami zapisanymi w tablicy. Jeśli któryś z warunków zostanie spełniony, program odpowiednio zmieni stan danego przekaźnika oraz diody LED ukazującej jego stan. Na końcu zerowana jest zmienna flagowa *Odebr_kod* oraz odblokowywane zostaje przerwanie INT0. Listing 2 (dostępny w Elportalu) ukazuje podprogram przerwania INT0, które zostaje wywołane stanem niskim na porcie podłączonym do odbiornika. Na samym początku zablokowane zostaje zewnętrzne przerwanie INT0, aby nie zostało wywołane powtórnie do czasu przetworzenia odebranej ramki. Kolejne instrukcje odblokowują globalny system przerwań i wywołują funkcję *Getrc5*, która używa przerwania od licznika Timer0 do generowania opóźnień podczas próbkowania sygnału. *Togglebit*, który jest zwracany przez funkcję w najstarszym bicie zmiennej *Odebrana_komenda*, zostaje przypisy do osobnej zmiennej i wyzerowany. Następne instrukcje sprawdzają, czy ramka została dobrze zdekodowana. Jeśli program napotkał problem, od razu odblokowywane jest przerwanie INT0. Kolejne instrukcje służą do sprawdzenia, czy przycisk nie został przytrzymany. Program porównuje odczytaną oraz poprzednią wartość zmiennych *Odebrana_komenda* oraz *Odebrany_togglebit*. Pomocniczo została użyta zmienna *Pierwsze_uruchomienie*, która zabezpiecza algorytm przy pierwszym uruchomieniu. Listing 3 (dostępny w Elportalu) przedstawia fragment podprogramu *Ustawienia*. Na samym początku wyłączane są wszystkie wyjścia mocy, po czym program wchodzi w pętlę, w której zmienia stan pierwszej diody co 100ms, aby poinformować użytkownika o konieczności wprowadzenia pierwszego przycisku. Po wyjściu z pętli odebrana komenda zostaje zapisana w tablicy. Następnie program wchodzi w pętlę, oczekując na potwierdzenie przycisku. W przypadku, gdy użytkownik nie potwierdzi wcześniej wciśniętego przycisku, program po upływie 5s wyjdzie z pętli i powróci na początek głównej pętli, aby powtórnie skalibrować przycisk. Takie rozwiązanie eliminuje zakłócenia i przypadkowe wciśnięcie



Rys. 3

przycisku. Pozostała część podprogramu działa analogicznie, z tym że porównuje odebrany kod z wcześniejszym, aby nie doszło do powtórzeń, a na samym końcu zapisuje ustawienia w pamięci EEPROM.

Montaż i uruchomienie

Schemat montażowy został przedstawiony na rysunku 3. Kolejność wlotowywania elementów nie jest obowiązkowa, ale warto zacząć od najmniejszych aż po największe. Podczas montażu należy pamiętać, aby wlotować wszystkie trzy zworki (pod i obok mikrokontrolera oraz obok odbiornika podczerwiieni). Ścieżki pomiędzy złączami śrubowymi a przekaźnikami najlepiej pokryć warstwą cyny, aby zapewnić lepszą wytrzymałość prądową. Pod mikrokontroler warto zastosować podstawkę DIP28. Następnie trzeba go zaprogramować. W tym celu należy wgrać plik *wsad.hex* do pamięci FLASH mikrokontrolera za pomocą dowolnego programatora dla mikrokontrolerów AVR, na przykład STK200/300 lub STK500. Fusebitów nie trzeba ustawiać, ponieważ układ opiera się na ustawieniach fabrycznych.

Zastosowane przekaźniki NT73-2C-S12 w razie potrzeby można zastąpić innymi, ważne jest, aby miały cewkę zasilaną napięciem 5V, te same wyprowadzenia oraz wymiary (na rysunku 4 zwymiarowano i opisano jeden z nich). Po wlotowaniu wszystkich elementów i zaprogramowaniu mikrokontrolera układ jest gotowy do

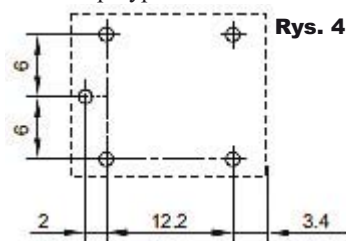
pracy. Powinien być zasilany napięciem stałym o wartości 5V. Przy pierwszym uruchomieniu należy zwrócić pierwszy przełącznik, aby wejść do ustawień i skalibrować pilot. Migająca dioda LED wskazuje przekaźnik, dla którego należy przypisać przycisk zmieniający jego stan. Po wciśnięciu przycisku należy po chwili go potwierdzić. Po pierwszym wciśnięciu dioda zaczyna świecić światłem stałym, natomiast jeśli przycisk nie zostanie potwierdzony, zaczyna powtórnie migać. Powyższe czynności należy powtórzyć dla pozostałych przekaźników, po czym należy wybrać przyciski, które będą wyłączać bądź włączać wszystkie przekaźniki.

Proces ten następuje analogicznie, ale należy pamiętać, że dioda D13 oraz D14 informują kolejno o kalibrowaniu przycisku odpowiednio wyłączającego i włączającego wszystkie przekaźniki. O poprawnie zakończonym procesie kalibracji informują włączone wszystkie diody LED. Aby wyjść z ustawień, należy rozewrzeć pierwszy przełącznik. Drugi przełącznik służy do awaryjnego wyłączenia wszystkich przekaźników bez konieczności stosowania pilota, przy czym blokuje także odbiór przez podczerwień, natomiast przełącznik trzeci blokuje odbiór przez podczerwień, nie zmieniając stanu przekaźników. Takie rozwiązanie może okazać się szczególnie przydatne, kiedy użytkownik po ustawieniu stanów wyjść mocy chce, aby nie zostały przypadkowo zmienione. Diody D7–D12 ukazują stan wyjść mocy, dioda D13 informuje o zablokowanym odbiorze przez podczerwień, natomiast dioda D14 o wyłączeniu wszystkich przekaźników i uniemożliwieniu wprowadzania zmian za pośrednictwem pilota.

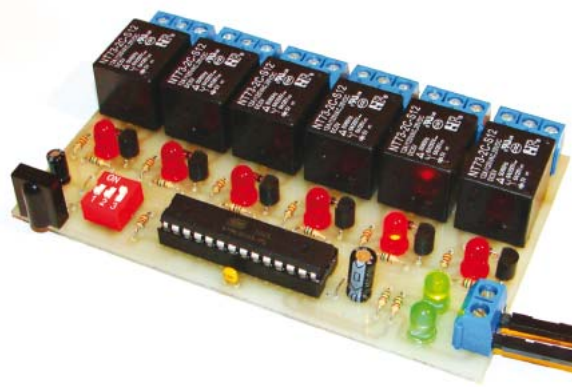
Wykaz elementów

R1	10kΩ
R2	220Ω
R3-R8	3,3kΩ
R9-R16	150Ω
C1	10μF/16V
C2	100nF
C3	4,7μF/16V
D1-D6	1N4148
D7-D14	LED 5mm
T1-T6	BC557
U1	ATmega8
U2	TSOP31236
K1-K6	NT73-2C-S12
S1	dip-switch x3
CON1	ARK500/2
CON2-CON7	ARK500/3

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-3053



Rys. 4



Krzysztof Gońka
krzysztof.gonka@interia.pl