

„2 W 1” czyli elektroniczna ruletka i kostka do gry



Do czego to służy?

Opisany moduł przeznaczony jest dla szerokiej rzeszy miłośników różnego rodzaju zabaw i gier, a w szczególności mających złyk hazardzisty. Jak sama nazwa wskazuje „2 w 1” jest to połączenie dwóch różnych projektów w jednym. Sięgając po wcześniejsze numery EdW, możemy odnaleźć kilka projektów dotyczących elektronicznej kostki i ruletki. Są także gotowe kity do samodzielnego montażu, ale nie było jeszcze projektu łączącego te dwie różne „zabawki”. Patrząc na fotografię modelu, ułożenie diod na płytce nie odzwierciedla oczek na kostce, ale jeśli zmontujecie sobie taki właśnie układ, zobaczycie, że łatwo się w tym połączyć. Mianowicie, liczba zapalonych diod będzie oznaczać liczbę oczek na kostce. Z ruletką nie powinno być najmniejszych problemów, jest klasyczna. A co powiecie, aby sterować tą zabawką pilotem na podczerwień? Myślę, że wszystkim znudziły się przyciski umieszczone bezpośrednio na płytce, a sterowanie bezprzewodowe jest teraz na topie. Do wykonania układu zachęcam wszystkich, którzy lubią się bawić!

Jak to działa?

Schemat ideowy układu pokazany jest na rysunku 1. Połączenie kostki i ruletki nie byłoby możliwe, gdyby nie mikrokontroler z rodziny AVR. Mowa oczywiście o AT90S2313, który jest bardzo popularny ostatnimi czasy w μProjektach 3000. Program znajdujący się w jego wnętrzu został napisany w Bascomie AVR ver.1.11.6.4. Jak widać, układ jest bardzo prościutki, zawiera dosłownie kilka elementów, tak więc nie powinien sprawić Wam żadnych problemów. Zajmijmy się teraz oprogramowaniem. Opuścimy deklarację zmiennych i konfigurację portów, a przejdziemy od razu do pętli głównej – listing 1. Jak widać, jest ona bardzo króciutka, zawiera cztery funkcje. Pierwsza funkcja *Rnd* oznacza skrót z języka angielskiego Random

Number Drawing, czyli generacja liczb losowych. Funkcja ta jest niewątpliwie rewelacją w Bascomie, bo nie musimy wymyślać żadnej procedury pozwalającej generować liczby losowe, tylko po prostu wpisujemy *Rnd* i w nawiasie górną wartość przedziału, z którego mają być generowane liczby. Druga funkcja *Getrc5* odpowiedzialna jest za dekodowanie kodów pochodzących z pilota, który pracuje oczywiście w standardzie RC5. Trzecia funkcja *Debounce* umożliwia podłączenie przełącznika bezpośrednio do linii portu oraz eliminuje efekt drgania jego styków. Zadaniem czwartej funkcji *Waitms* jest spowolnienie pracy trzech wcześniejszych. Oprócz pętli głównej w programie znajdują się dwie procedury: *Kostka* i *Ruletka*. W zależności od zmiennej *Kod*, która przyjmuje war-

tości zdekodowanego kodu, program wykonuje odpowiednią procedurę. Naciśnięcie klawisza o kodzie 1 lub *Przycisku_1* (S1) spowoduje wywołanie procedury *Kostka* – listing 2. Na początku procedury umieściłem w pętli powtórzeń „symulację” losowania,

Listing 1

```

Do
Liczba_generowana = Rnd(6)
Waitms 100
Getrc5(address , Kod)          'dekodowanie kodów
If Address = 0 Or Address = 10 Then
    Kod = Kod And &B10111111
    End If

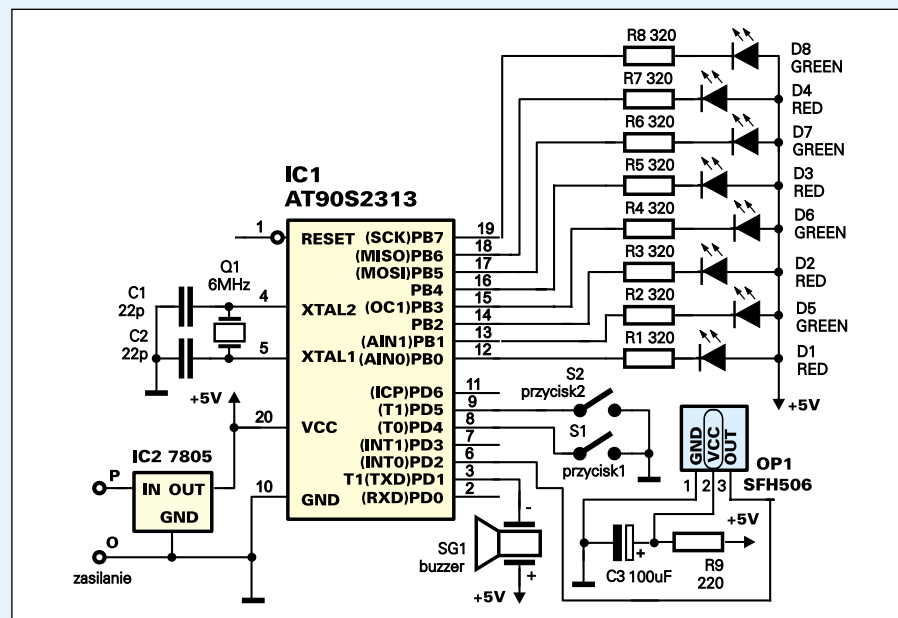
If Kod = 1 Then
    Call Kostka
    End If

If Kod = 2 Then
    Call Ruletka
    End If

Debounce Przycisk_1 , 0 , Kostka , Sub
Debounce Przycisk_2 , 0 , Ruletka , Sub

Loop
    
```

Rys. 1 Schemat ideowy



polegającą na miganiu sześciu diod z towarzyszącym im dźwiękiem. Następnie w zależności od wygenerowanej w pętli głównej liczby ustawić port w stan odzwierciedlający tę liczbę. Zastosowałem do tej operacji funkcję *Select Case*. Wynik w postaci zapalonych diod widoczny jest przez 5 sekund, a po upływie tego czasu diody gasną.

Listing 2

```
Sub Kostka:
    For L = 1 To 15
        Diody = &B11000000
        Sound Bzyczek , 500 , 50
        Diody = &B11111111
        Waitms 150
        Set Bzyczek
        Next L
    Select Case Liczba_generowana
    Case 0 : Diody = &B11111110
    Case 1 : Diody = &B11111100
    Case 2 : Diody = &B11111000
    Case 3 : Diody = &B11110000
    Case 4 : Diody = &B11100000
    Case 5 : Diody = &B11000000
    End Select
    Wait 5
    Diody = &B11111111
    End Sub Kostka
```

Zasadę działania elektronicznej *Kostki* mamy już za sobą, przejdziemy więc do wyjaśnienia, na czym będzie polegać działanie *Ruletki* – listing 3. Wejście do tej procedury możliwe jest po naciśnięciu klawisza o kodzie 2 lub *Przycisku_2* (S2). Jak na pewno zauważyliście, jest tu pełno, chętnie przeze mnie stosowanych, pętli powtórzeń „wzbogacających” o funkcję *Rotate*. Na początku procedury zakładam początkową wartość zmiennej

Listing 3

```
Sub Ruletka:
    Rolowana_liczba = &B11111110
    For L = 1 To 25
        Zwloka = Zwloka + 5
        Rotate Rolowana_liczba , Left , 1
        Diody = Rolowana_liczba
        Sound Bzyczek , 800 , 10
        Set Bzyczek
        For I = 1 To Zwloka
            Waitms 2
            Next I
        Next L
        For L = 1 To Liczba_generowana
            Zwloka = Zwloka + 5
            Rotate Rolowana_liczba , Left , 1
            Diody = Rolowana_liczba
            Sound Bzyczek , 800 , 10
            Set Bzyczek
            For I = 1 To Zwloka
                Waitms 2
                Next I
            Next L
        Next L
    Wait 5
    Diody = &B11111111
    End Sub Ruletka
```

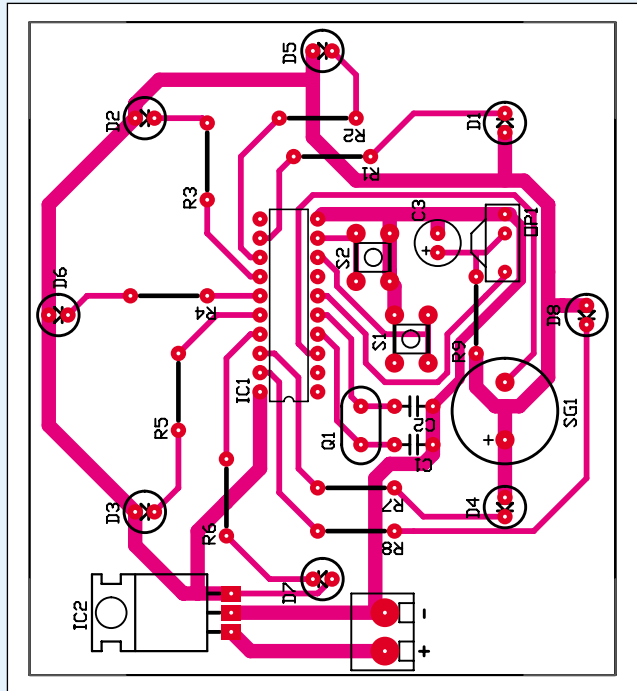
A, którą będę obracał o jedno miejsce w lewo. Rolowaniu będzie towarzyszył dźwięk. Zastosowanie jednej pętli w drugiej pozwoliło na stopniowe zmniejszanie prędkości obracania bitu. Aby „kulka” nie zatrzymywała się w jednym miejscu, powtórzyłem drugi raz tę samą pętlę, tylko liczbę powtórzeń zależy od zmiennej *Liczba_generowana*, czyli wygenerowanej w pętli głównej liczby. Tak jak w przypadku *Kostki* wynik widoczny jest przez pewien czas. Oprogramowanie mamy za sobą, w razie niejasności polecam analizę kodu źródłowego dostępnego na stronie internetowej EdW.

Montaż i uruchomienie

Cały układ można zamontować na płytce drukowanej pokazanej na rysunku 2. Z własnego doświadczenia polecam rozpoczęcie montażu od zamontowania stabilizatora i sprawdzenia poprawności jego działania. „Zaszycie” całej inteligencji układu we wnętrzu mikroprocesora spowodowało, że montaż płytki jest banalny, więc przeprowadzamy go w sposób typowy, od elementów najniższych do najwyższych. Diody LED montujemy na przemian raz koloru czerwonego, raz koloru zielonego. Dwa kolory mają odzwierciedlać klasyczną ruletkę. Niestety nie ma diod świecących na czarno, dlatego użyłem koloru zielonego. Po wlutowaniu podstawki i umieszczeniu w niej zaprogramowanego mikroprocesora możemy zaczynać zabawę.

Obsługa układu

Przy okazji omawiania programu wspominałem już o obsłudze naszej „zabawki”, ale chcę tę informację zebrać osobno. Obsługa jest banalnie prosta: naciskając klawisz 1 lub przycisk 1 (S1) „rzucamy kostką”, naciskając klawisz 2 lub Przycisk 2 (S2) „puszczamy kuleczkę”. Życzę wszystkim miłej zabawy!!!



Rys. 2 Schemat montażowy

Wykaz elementów

Rezystory

- R1-R8320Ω
- R9220Ω

Kondensatory

- C1,C222pF
- C3100μF

Półprzewodniki

- D1-D4LED5 czerwone
- D5-D8LED5 zielone
- IC1AT90S2313
- IC27805
- OP1SFH506

Inne

- Q18MHz
- SG1Buzzer piezo
- ARK 2

Komplet podzespołów z płytką jest dostępny w sieci handlowej AVT jako kit szkolny AVT-3019

Łukasz Hrapek