

# BASIC Stamp

## w roli elektronicznej kostki do gry

Po kilku artykułach prezentujących możliwości komputerdka BASIC Stamp i zaimplementowanego w nim języka, przedstawiamy przykład prostej, lecz bardzo efektownej aplikacji. Jest to mikroprocesorowa kostka do gry (podwójna!) z wyświetlaczem wyniku w postaci 14 diod LED, charakteryzująca się minimalną liczbą elementów zewnętrznych.

### Listing 1

' Program KOSTKA.BAS  
' Podwójna elektroniczna kostka do gry

```
Symbol      kostka1 = b0
Symbol      kostka2 = b1
Symbol      losowanie = w3
Symbol      oczka = b2
Symbol      Wybierz = 7
```

```
high Wybierz
let dirs = 255
let kostka1 = 11111
let kostka2 = 41111
```

```
Powtorz:
let oczka = kostka1
gosub Wskaznik
let oczka = kostka2
gosub Wskaznik
input Wybierz
if pin7 = 1 then Losuj
let w3 = w3+1
Reenter:
output Wybierz
goto Powtorz
```

```
Wskaznik:
lookup oczka, (64,18,82,27,91,63), oczka
let pins = pins&10000000
toggle Wybierz
let pins = pins|oczka
pause 41111111111111111111
return
```

```
Losuj:
random losowanie
let kostka1 = b6&*00000111
let kostka2 = b7&*00000111
if kostka1 > 5 then Losuj
if kostka2 > 5 then Losuj
goto Reenter
```

Na rys. 1 znajduje się schemat elektryczny proponowanego rozwiązania mikroprocesorowej, podwójnej kostki do gry. Każda z tych dwóch kostek jest reprezentowana przez układ siedmiu diod LED, których zapalenie sygnalizuje liczbę „wyrzuconych“ oczek. Łatwo zauważyć, że diody świecące są sterowane bezpośrednio z portu BASIC Stampa. Taka konfiguracja układu jest prawidłowa, ponieważ wydajność prądowa każdej linii portu sięga 20mA.

Linia P7 steruje przełącznikiem zbudowanym na tranzystorach T1 i T2, który kluczuje grupy diod tworzących kostki. Mamy więc do czynienia z sekwencyjnym wyświetlaniem wyniku losowania. Dodatkową funkcją linii P7 jest obsługa przycisku losowania.

Zasada pracy kostki opiera się na generowaniu przypadkowych ciągów liczb, w oparciu o wbudowaną w PBASIC funkcję generatora ciągu pseudolosowego.

Na list. 1 przedstawiono program podwójnej kostki do gry, napisany w języku PBASIC. Zadaniem procedury głównej jest sekwencyjne wyświetlanie wyniku losowania oraz wykrywanie naciśnięcia przycisku losowania SW1.

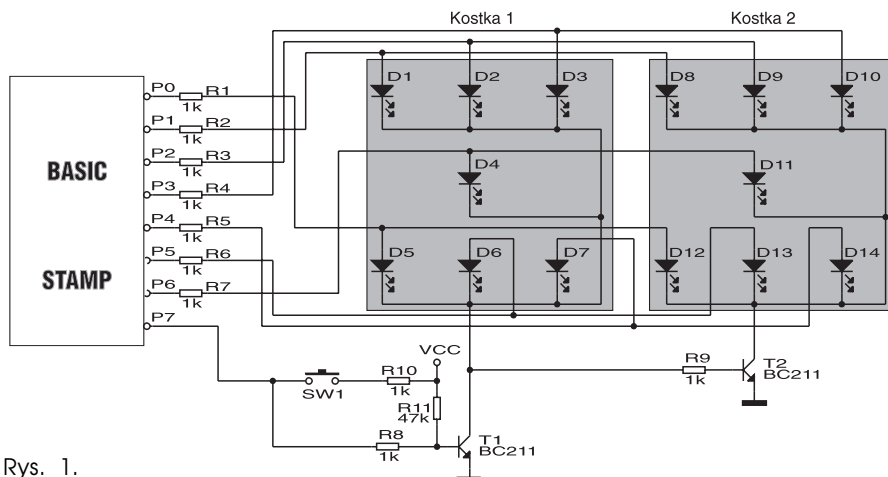
Po wykryciu naciśnięcia przycisku SW1, program przechodzi do realizacji podprogramu Losuj. Przy jego pomocy zostaje wytworzona 16-bitowa liczba pseudolo-

sowa, umieszczona w słowie w3 (zastępcza nazwa losowanie). Liczba ta będzie podstawą dla tworzenia wyniku losowania. Jak wiadomo, słowo w3 odpowiada w strukturze danych PBASICA bajtom b6 i b7. Wartości tam zapisane, po wymaskowaniu trzech najmłodszych bitów i wyzerowaniu pozostałych bitów, są umieszczone odpowiednio w zmiennych kostka1 i kostka2. Oczywiście, interesuje nas tylko sześć różnych liczb, a nie osiem, bowiem tyle będzie dostępnych po tej operacji. Pozostaje więc jeszcze ograniczyć ten zakres liczb poprzez odrzucenie liczb większych od 5. W przypadku zapisu do kostka1 albo kostka2 co najmniej jednej liczby większej od 5, losowanie jest powtórzone.

Wyświetlaniem zajmuje się podprogram Wskaznik. Podprogram ten dokonuje przekodowania z zapisu binarnego na układ świecących diod LED. Bardzo pomocnym rozkazem jest LOOKUP. Zwraca on jedną z wartości zawartą w nawiasie o takiej liczbie porządkowej, jaka jest zapisana w pierwszym argumencie. Zwracana wartość jest umieszczana w trzecim argumencie instrukcji. W ten sposób bardzo łatwo w języku PBASIC tablicować różne funkcje.

Następne operacje tego podprogramu są ściśle związane z wyświetlaniem informacji. Najpierw wszystkie diody LED są wygaszone (operacja zerowania pinów P6..P0, stan pinu P7 pozostaje nie zmieniony), potem jest zmieniony stan pinu P7 na przeciwny, następnie jest ustawiony właściwy układ włączonych diod LED. Taka kolejność działań pozwala uniknąć „duchów“, czyli słabo świecących stanów drugiej kostki na diodach kostki pierwszej.

Mirosław Lach, AVT



Rys. 1.