

# Cyfrowa kostka do gry

## Do czego to służy?

Prezentowany układ pełni funkcję typowej sześcienniej kostki do gry. Dzięki zastosowaniu siedmiosegmentowego wyświetlacza LED oraz przycisku typu microswitch, można dużo szybciej dokonywać losowań liczb 1...6 niż w przypadku zwykłej kostki. Podczas losowania, na wyświetlaczu jest prezentowany wąż świetlny, co zwiększa efekt wizualny. Cały układ został zbudowany na małej płytce, jest zasilany napięciem 5V i pobiera zaledwie około 22mA.

## Jak to działa?

Na **rysunku 1** przedstawiono schemat ideowy. Układ sterowany jest mikrokontrolerem AVR ATtiny2313. Przycisk S1 podczas losowania zwierza PortD.1 z masą. Kondensatory C1 oraz C2 filtrują napięcia dla mikrokontrolera. Segmenty wyświetlacza zostały podłączone przez rezystory ograniczające prąd do portów mikrokontrolera. Na **listingu 1** została ukazana część odpowiedzialna za obsługę przycisku. Po jego kliknięciu, program czyści wyświetlacz oraz odblokowuje przerwanie od Timer0, w którym losowane są liczby. Następnie wchodzi w nieskończoną pętlę, z której może wyjść tylko wtedy, gdy przycisk zostanie zwolniony. Program przed opuszczeniem pętli blokuje przerwanie od Timer0. Po wyjściu z pętli zostaje czyszczony wyświetlacz oraz następuje opóźnienie 100ms, po czym pobierany jest kod wylosowanej cyfry i zapisywany na wyświetlaczu. Wnętrze przerwania, w którym następuje losowanie liczby, ukazuje **listing 2**. Przerwanie to jest zgłaszane co 100us, dzięki czemu algorytm zapewnia dobrą losowość. Pierwszą instrukcją jest załadowanie licznika Timer0 nową wartością 100. Następnie zachodzi inkrementacja zmiennej Wylosowana. Kolejna instruk-

cja po przekroczeniu zakresu losowanych liczb ustawia wartość na 0, dzięki czemu układ ciągle liczy od 0 do 5. Pozostałe instrukcje przerwania służą do wyświetlania węża świetlnego na wyświetlaczu w trakcie losowania.

## Montaż i uruchomienie

Na **rysunku 2** przedstawiono schemat montażowy. Kolejność wlotowywania elementów nie ma większego znaczenia, ale warto rozpocząć od elementów najmniejszych aż po największe. Zastosowany wyświetlacz LSD052BSR-10 w razie potrzeby można zastąpić innym ze wspólną anodą. Pod mikrokontroler najlepiej zastosować podstawkę DIP20. Następnie trzeba go zaprogramować. W tym celu należy do pamięci Flash mikrokontrolera wgrać dostępny w Elportalu wśród materiałów dodatkowych tego numeru, plik *wsad.hex*. Można to zrobić za pomocą dowolnego programatora dla mikrokontrolerów AVR, na przykład STK200/300 lub STK500. Fusebitów nie trzeba ustawiać,

ponieważ układ opiera się na ustawieniach fabrycznych.

Obsługa układu jest łatwa i opiera się na jednym przycisku typu microswitch. Aby wylosować liczbę, należy przytrzymać przez chwilę przycisk. Wylosowana liczba jest uzależniona od długości przytrzymania przycisku, ale nie jest możliwe, aby użytkownik przewidywał losowane liczby, gdyż mikrokontroler robi to z częstotliwością 10kHz.

Aby zmienić domyślny zakres losowanych liczb w granicach 0...9, należy w programie zmienić wartość maksymalną w instrukcji zerującej zmienną przechowującą wylosowaną wartość.

## Wykaz elementów

R1-R7	.....	150Ω
C1	.....	10μF/16V
C2	.....	100nF
U1	.....	ATtiny2313
W1	.....	LSD052BSR-10
S1	.....	przycisk typu microswitch
CON1	.....	łącze śrubowe ARK500/2

**Komplet podzespołów z płytka jest dostępny w sieci handlowej AVT jako kit szkolny AVT-2998.**

Krzysztof Gońka

krzysztof.gonka@interia.pl

Rys. 2

```

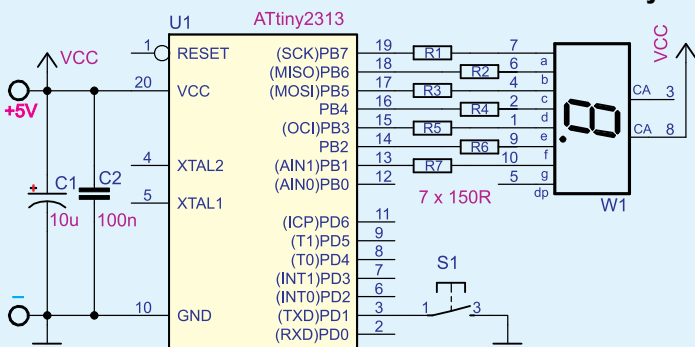
Do
  If S1 = 0 Then
    Portb = 255
    Enable Timer0
  Do
    If S1 = 1 Then
      Disable Timer0
      Exit Do
    End If
  Loop
  Portb = 255
  Waitms 100
  Stan_portu = Lookup(Wylosowana, Wyświetlacz)
  Portb = Stan_portu
End If
Loop
End
  
```

**Listing 1**

```

Przerwanie:
Load Timer0 , 100
Incr Wylosowana
If Wylosowana = 6 Then Wylosowana = 0
Incr Licznik
If Licznik = 255 Then
  Licznik = 0
  Incr Kreska
  If Kreska = 8 Then
    Portb.7 = 1
    Kreska = 2
  End If
  Portb.kreska = 0
  Kreska = Kreska - 1
  Portb.kreska = 1
  Kreska = Kreska + 1
End If
Return
  
```

**Listing 2**



Rys. 1

