Electronic Dice

Design Brief

Design an electronic version of a dice.

Electronic Circuit

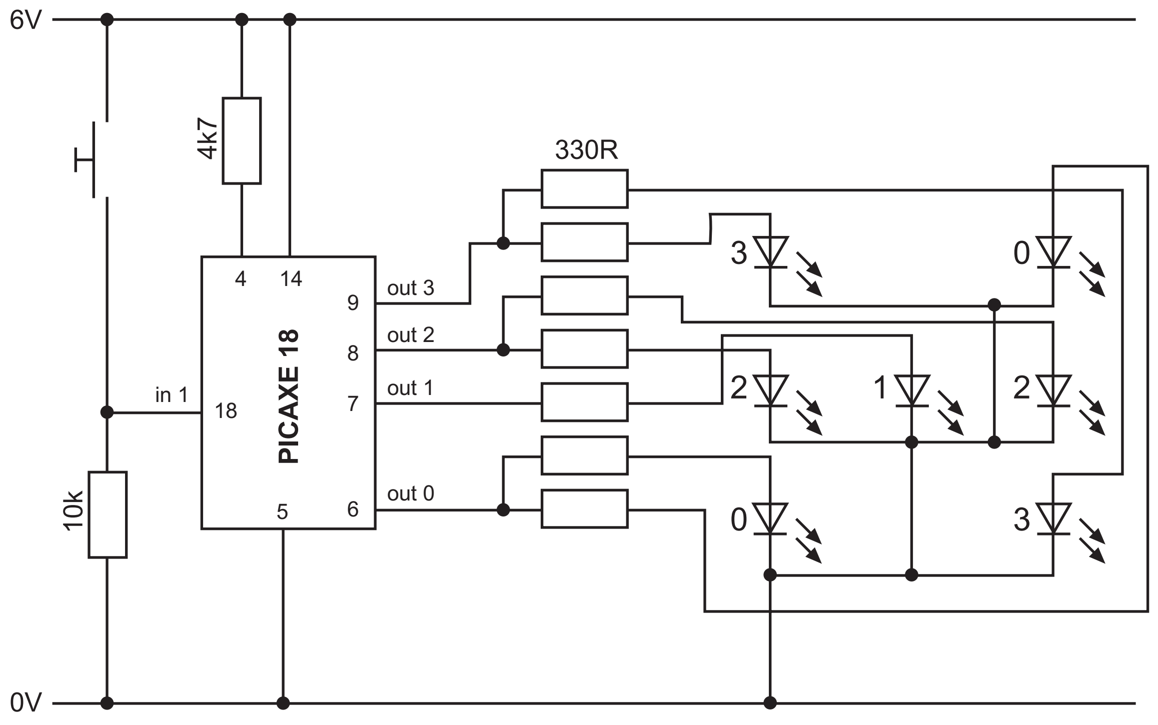
The electronic circuit shows LEDs connected to outputs 0 to 3 to make a single die. This circuit could be repeated on outputs 4 to 7 to give a pair of dice. Note that the LEDs are paired up in opposite corners as they only light in pairs, and so it would be a waste of output pins to connect each individually.

Program Explanation

Two programs are given – one for a single die and the other for a pair of dice. Both programs use the random command to generate a random byte number between 0 and 255. This number is then tested to give a value 1 to 6 for the die, and the 'let pins =' command sets the correct output display for the LEDs.

Note that with a microcontroller, the random command will always generate the same pattern of numbers due to the way it internally processes numbers. Therefore by putting the random command within the 'wait for switch press' loop, the pattern is disrupted (to give a truly random number) as there will be varying time delays between switch pushes.

The second program uses the random command to generate random numbers in word variable w0. A word variable uses two bytes, and so w0 consists of the two byte variable b0 and b1. Therefore after the command 'random w0 'both b0 and b1 will contain random numbers – one for each of the two dice. The two dice patterns are then combined by an OR (|) command before the 'let pins = ' statement.



Program Listing

```
' Dice Game (Single Die)
' For PICAXE-18

' switch on input 1
' output0 = diagonal /
' output1 = centre dot
' output2 = centre -
' output3 = diagonal \

' 3   0
' 2 1 2
' 0   3

' loop that gets random values
' into variable b0
' also shows varying values on LEDs
' to give impression LEDs are rolling
' when switch is pressed jump down to display

main: if pin1 = 1 then display
      random b0
      let pins = b0
      goto main

' check random value and split into
' six equal parts to get digits
' 1 to 6 from possible values 0 to 255

display:
      if b0 > 215 then show6
      if b0 > 172 then show5
      if b0 > 129 then show4
      if b0 > 86 then show3
      if b0 > 43 then show2
      goto show1

show6:     let pins = %00001101
      pause 2000
      goto main

show5:     let pins = %00001011
      pause 2000
      goto main

show4:     let pins = %00001001
      pause 2000
      goto main

show3:     let pins = %00001010
      pause 2000
      goto main

show2:     let pins = %00000100
      pause 2000
      goto main

show1:     let pins = %00000010
      pause 2000
      goto main
```

```
‘ Dice Game (Double Dice)
‘ For PICAXE-18

‘ Note the lines marked ‘*** were unnecessary
‘ and so were commented out as the original
‘ program was too long for the PICAXE-18 chip!
‘(PICAXE-28 has twice the memory and so would be ok)

‘ switch on input 1
‘ output0 = 1 diagonal /
‘ output1 = 1 centre dot
‘ output2 = 1 centre -
‘ output3 = 1 diagonal \
‘ output4 = 2 diagonal /
‘ output5 = 2 centre dot
‘ output6 = 2 centre -
‘ output7 = 2 diagonal \

‘ 3   0 7   4
‘ 2 1 2 6 5 6
‘ 0   3 4   7

‘ loop that gets random values
‘ into word w0 (=b0 and b1)
‘ also shows varying values on LEDs
‘ to give impression LEDs are rolling
‘ when switch is pressed jump down to display

main: if pin1 = 1 then display
      random w0
      let pins = b0
      goto main

‘ check random value of b0 and split into
‘ six equal parts to get digits
‘ 1 to 6 from possible values 0 to 255
‘ store the value in variable b3 for time being

display:
      if b0 > 215 then show6
      if b0 > 172 then show5
      if b0 > 129 then show4
      if b0 > 86 then show3
      if b0 > 43 then show2
‘***  goto show1

show1:     let b3 = %00000010
      goto display2

show2:     let b3 = %00000100
      goto display2

show3:     let b3 = %00001010
      goto display2

show4:     let b3 = %00001001
      goto display2

show5:     let b3 = %00001011
      goto display2

show6:     let b3 = %00001101
‘***  goto display2
```

```
' Now get second dice value from
' variable b1 and OR into top half of b3
' to combine the two dice patterns together

display2:
     if b1 > 215 then show6a
     if b1 > 172 then show5a
     if b1 > 129 then show4a
     if b1 > 86 then show3a
     if b1 > 43 then show2a
'*** goto show1a

show1a:    let b3 = b3 | %00100000
     goto display3

show2a:    let b3 = b3 | %01000000
     goto display3

show3a:    let b3 = b3 | %10100000
     goto display3

show4a:    let b3 = b3 | %10010000
     goto display3

show5a:    let b3 = b3 | %10110000
     goto display3

show6a:    let b3 = b3 | %11010000
'*** goto display3

' Now actually display on the LEDs

display3:
     let pins = b3
     pause 2000
     goto main
```