

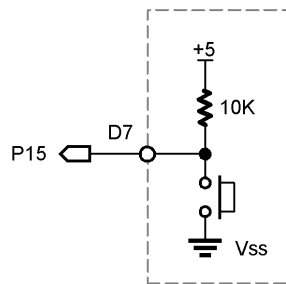


Experiment #9: A Digital Die

The purpose of this experiment is create a digital die (one half of a pair of dice).

Building The Circuit.

Add this pushbutton to the circuit in Experiment #8.



```

=====
'
'
'   File..... Ex09 - Roller.BS2
'   Purpose... Digital Die
'   Author.... Parallax
'   E-mail.... stamptech@parallaxinc.com
'   Started...
'   Updated... 01 MAY 2002
'
'   {$STAMP BS2}
'
=====
'
'-----
' Program Description
'-----
'
' This program combines a 7-segment display and a pushbutton input to create
' a single-digit digital die. Displays 1 to 6 when button is pressed.
'

```

Experiment #9: A Digital Die

```
'-----
' I/O Definitions
'-----

RollBtn      CON      15          ' roll button on Pin 15
Segs         VAR      OutL        ' 7-segment LEDs

'-----
' Variables
'-----

swData       VAR      Byte          ' data for BUTTON command
dieVal       VAR      Nib           ' new die value
spinPos      VAR      Nib           ' spinner position
doSpin       VAR      Nib           ' spinner update control

'-----
' EEPROM Data
'-----

'                abcdefg
'                -----
DecDig       DATA    %01111110     ' 0
              DATA    %00110000     ' 1
              DATA    %01101101     ' 2
              DATA    %01111001     ' 3
              DATA    %00110011     ' 4
              DATA    %01011011     ' 5
              DATA    %01011111     ' 6
              DATA    %01110000     ' 7
              DATA    %01111111     ' 8
              DATA    %01111011     ' 9

Bug          DATA    %01000000     ' spinning bug
              DATA    %00100000
              DATA    %00010000
              DATA    %00001000
              DATA    %00000100
              DATA    %00000010

'-----
' Initialization
'-----
```

```
Initialize:
  DirL = %01111111      ' create output pins

' -----
' Program Code
' -----

Main:
  GOSUB Get_Die          ' update die value
  PAUSE 5
  ' is the button pressed?
  BUTTON RollBtn, 0, 255, 10, swData, 1, Show_Die
  GOTO Main              ' no

Show_Die:
  READ (DecDig + dieVal), Segs      ' show the die
  PAUSE 3000                        ' - for 3 seconds
  GOTO Main                          ' go again

  END

' -----
' Subroutines
' -----

Get_Die:
  dieVal = (dieVal // 6) + 1      ' limit = 1 to 6
  READ (Bug + spinPos), segs      ' show spinner pattern
  doSpin = (doSpin + 1) // 7      ' time to update spinner?
  IF (doSpin > 0) THEN Get_DieX   ' only if doSpin = 0
  spinPos = spinPos + 1 // 6      ' update spinner

Get_DieX:
  RETURN
```

Behind The Scenes

This program borrows heavily from what we've already done and should be easy for you to understand. What we've done here is added a bit of programming creativity to make a very simple program visually interesting.

Experiment #9: A Digital Die

There is one noteworthy point: the use of the variable, `doSpin`. In order to create a random value, the variable `dieVal` is updated rapidly until the button is pressed. This rate of change, however, is too fast to allow for a meaningful display of the rotating “bug.” The variable `doSpin`, then, acts as a delay timer, causing the LED “bug” position to be updated every seventh pass through the `Get_Die` routine. This allows us to see it clearly and creates an inviting display.