

# Electronic Dice

In this project, you will build on the Rainbow LED project with the TOP by turning it into a dice. By using the random function to generate a number and assigning each LED a value (all 5 lit up represents six), you can make a very simple electronic dice.

## What You Need:

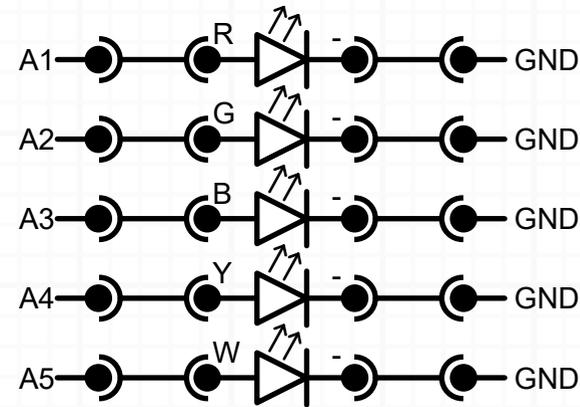
1x	JackBord	1x	JackBord TOP
10x	Female-Female Jumper		

Note: It is assumed you have already connected the TOP to the JackBord. If not, please refer to the **Using the JackBord TOP** guide, or activity **1001-act5**.

## Construction Steps:

- Using five jumper wires, connect PORT pins A1 through A5 to a corresponding TOP LED pin. So A1 to TOP LED pin R, A2 to TOP LED pin G, A3 to TOP LED pin B etc.
- Open the PROG page from the dashboard and enter the program shown on page 3.
- If you have made the Colour Night Light, you may notice that the primary **ant** statement is very similar. They are, and the only real change is that instead of time, the value compared is random.
- Once done, run the program and a random result should be generated and displayed on the LEDs.

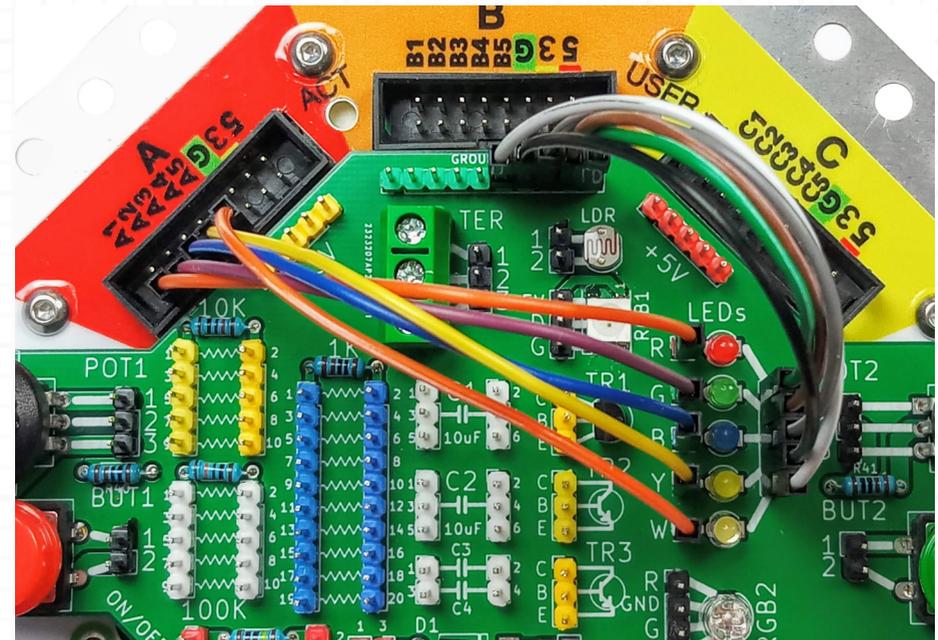
## Circuit Diagram



Port A pin 1	TOP LED pin R
Port A pin 2	TOP LED pin G
Port A pin 3	TOP LED pin B
Port A pin 4	TOP LED pin Y
Port A pin 5	TOP LED pin W
TOP LED pin -	TOP Ground
TOP LED pin -	TOP Ground
TOP LED pin -	TOP Ground
TOP LED pin -	TOP Ground
TOP LED pin -	TOP Ground

The table above contains the connections in the circuit diagram. Simply connect a jumper from the left column pins to the corresponding right column pin in the same row.

## Completed circuit



## Code:

```
---
Dice Program Default
---

prog_vars =
  d/count 0
prog_vars.

prog_start =
  pclear
  print "Program Start"
prog_start.

prog_loop =
  -- main program loop
  rand 1 6 -> /count
  print "Count: /count"
  ant /count ->
    = 1 -> a1 1
    = 2 -> a2 1
    = 3 -> a3 1
    = 4 -> a4 1
    = 5 -> a5 1
    = 6 -> a1 1|a2 1|a3 1|a4 1|a5 1
    -> print "wait"
  ant.
  exitprog
prog_loop.

prog_stop =
  dly 500
  a1 0|a2 0|a3 0|a4 0|a5 0
  print "Bye"
prog_stop.
```

## Code Ver 2:

```
---
Dice Program With Cheating
---

prog_vars =
  d/count 0
  d/roll 0
  d/rig 0
prog_vars.

prog_start =
  pclear
  print "Program Start"
  a1 0|a2 0|a3 0|a4 0|a5 0
prog_start.

prog_loop =
  -- main program loop
  btp b1 /roll 1
  btp b2 /roll 2
  any /roll = 1 ->
    repeat 2 ->
      rand 1 6 -> /count
      print "Count: /count"
      ant /count ->
        = 1 -> a1 1
        = 2 -> a2 1
        = 3 -> a3 1
        = 4 -> a4 1
        = 5 -> a5 1
        = 6 -> a1 1|a2 1|a3 1|a4 1|a5 1
        -> print "wait"
      ant.
      dly 500
      a1 0|a2 0|a3 0|a4 0|a5 0
    repeat.
    /roll 0
  enda.

  any /roll = 2 ->
    repeat 2 ->
      rand 1 6 -> /count
      rand 1 2 -> /rig
      any /rig = 1 ->
        dec /count
```

**Continued on next page**

```

enda.
print "Count: /count"
ant /count ->
    = 1 -> a1 1
    = 2 -> a2 1
    = 3 -> a3 1
    = 4 -> a4 1
    = 5 -> a5 1
    = 6 -> a1 1|a2 1|a3 1|a4 1|a5 1
    -> a1 1
ant.
dly 500
a1 0|a2 0|a3 0|a4 0|a5 0
repeat.
/roll 0
enda.
prog_loop.

prog_stop =

    print "Bye"
prog_stop.

```

## Next Steps:

---

Now that you have your dice 'rolling', let's take it to the next level. Instead of having to run the program each time, what about implementing a button instead? Instead of just a 'single' dice, what about having it roll twice to represent two dice? And seeing as this is dice, it's a time honoured tradition to see if you can rig them in your favour.

Let's start with adding some buttons. Connect BUT1 to pin b1 and BUT2 to pin b2. These will be the roll triggers for player 1 and 2. To get them to trigger the roll, add a roll variable that increases when the **btp** command detects a button press. Then have an any statement to detect and reset the value after performing a roll. Repeat this in a separate loop for the second button.

There are several ways to rig the roll, and you are free to try your own. In the example shown, the rigging is done by adding a 50% chance to decrease the roll of player 2 by 1 each roll.