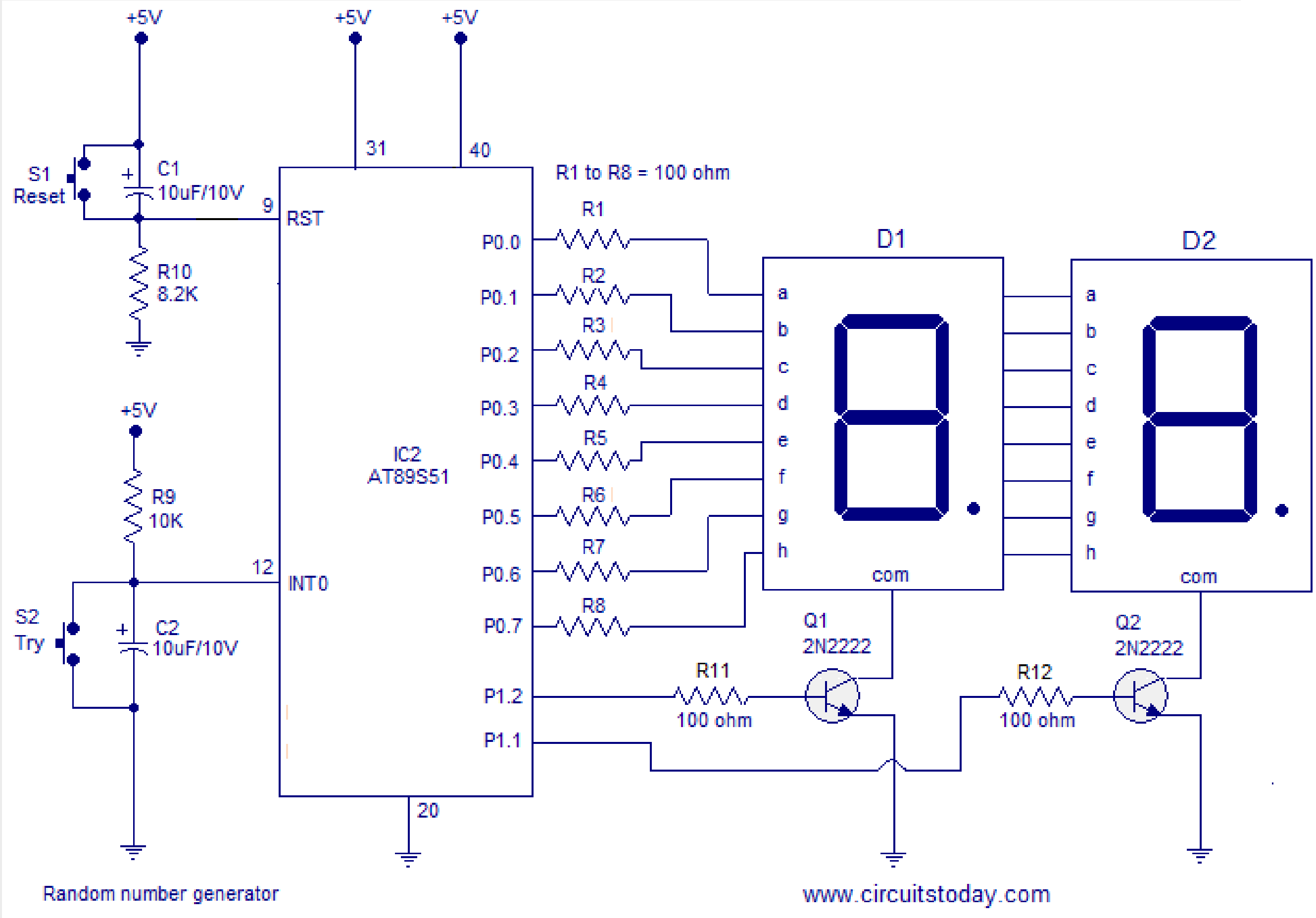
RANDOM NUMBER GENERATOR USING 8051

A random number generator using 8051 that displays a random number between 0 & 99 is shown in this article. The circuit it self is very simple and may not find any applications in serious embedded projects and this article is just an illustration. The circuit is based on AT89S51 microcontroller, two seven segment LED displays, two transistors and few passive components.

Circuit diagram.



Random number generator using 8051

The two seven segment LED displays are multiplexed together and their data lines are connected to Port0 of the microcontroller. Transistors Q1 and Q2 drives the corresponding displays D1 and D2. The driving signals for there transistors are obtained from P1.1 and P1.2. Push button switch S1,capacitor C1 and resistor R10 forms a debouncing reset circuit. Resistor R9, capacitor C2 and pushbutton switch S2 will provide an active low harware interrupt signal at INTO (pin12) when ever S2 is pressed. Here also R9 and C2 are meant for debouncing. After power ON the display will show blank and when push button S2 is pressed the display will show a random number between 0 and 99. For another try you have to press the reset switch and then switch S2. If you need a single digit setup only, the remove display D2 and its associated components. Everything else is same.

Program.

```
ORG 000H
SJMP MAIN
ORG 003H // sets the starting address for the ISR
ACALL ISR // calls the ISR subroutine when S2 is pressed
RETI // return from interrrupt

MAIN:SETB IP.0 // this part sets the initial conditions
     SETB TCON.0
     SETB IE.0
     SETB IE.7
     MOV P0,#00000000B
     MOV P1,#00000000B
     MOV DPTR,#LUT // moves the starting address of LUT to DPTR

LABEL:MOV R6,#99D // this part generates the random number
      LOOP:MOV A,R6
      DJNZ R6,LOOP
      SJMP LABEL

ISR: MOV A,R6 // Subroutine ISR displays the current random number
     MOV B,#10D
     DIV AB
     SETB P1.2
     ACALL DISPLAY
     MOV P0,A
     ACALL DELAY
     MOV A,B
     CLR P1.2
     SETB P1.1
     ACALL DISPLAY
     MOV P0,A
     ACALL DELAY
     CLR P1.1
     SJMP ISR
     RET

DELAY: MOV R3,#02H // this subroutine creates 1mS delay for
switching the displays
DEL1: MOV R2,#0FAH
DEL2: DJNZ R2,DEL2
      DJNZ R3,DEL1
      RET

DISPLAY: MOVC A,@A+DPTR // produces the digit drive pattern for the
current digit in A
         RET

LUT: DB 3FH // Look up table
     DB 06H
     DB 5BH
     DB 4FH
     DB 66H
     DB 6DH
     DB 7DH
     DB 07H
     DB 7FH
```

```
    DB 6FH
    END
```

About the program.

The first part of the program is the portion labelled MAIN which sets the initial conditions and the interrupt parameters. The next part is the loop named LABEL which loads 99D to register R6 then decrements it by 1 until 0 and then repeats the cycle again. This is the part which generates the random number. Every time R6 is decremented the resultant value is moved to accumulator A. Next part is the interrupt service routine which is written as a subroutine named ISR. When ever there is an interrupt at INT0 (push button S2 is pressed), the ISR is called. The ISR perfoms necessary mathematical manipulations on the content of A in order to split out the two digits and then proceeds to show it on the display. Subroutine DELAY produces roughly 1ms delay for switching the displays. Subroutine DISPLAY adds the current value in A with the address stored in DPTR (starting address of LUT) and moves the target content to A. The result will be the digit drive pattern for the current digit in A.

Source : http://www.circuitstoday.com/random-number-generator-using-8051