

DT-51

DT-51 *Application Note*

AN35 – Dadu Elektronik

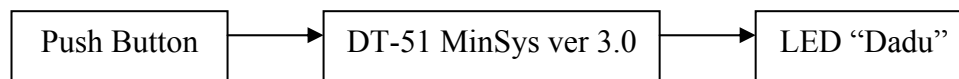
Oleh: Tim IE & Ary Mulyono
(Universitas Katholik Widya Mandala)

Aplikasi ini berfungsi menyerupai dadu. Dengan kemampuan mengacak (*random*), nilai kedua “dadu” sulit untuk ditebak.

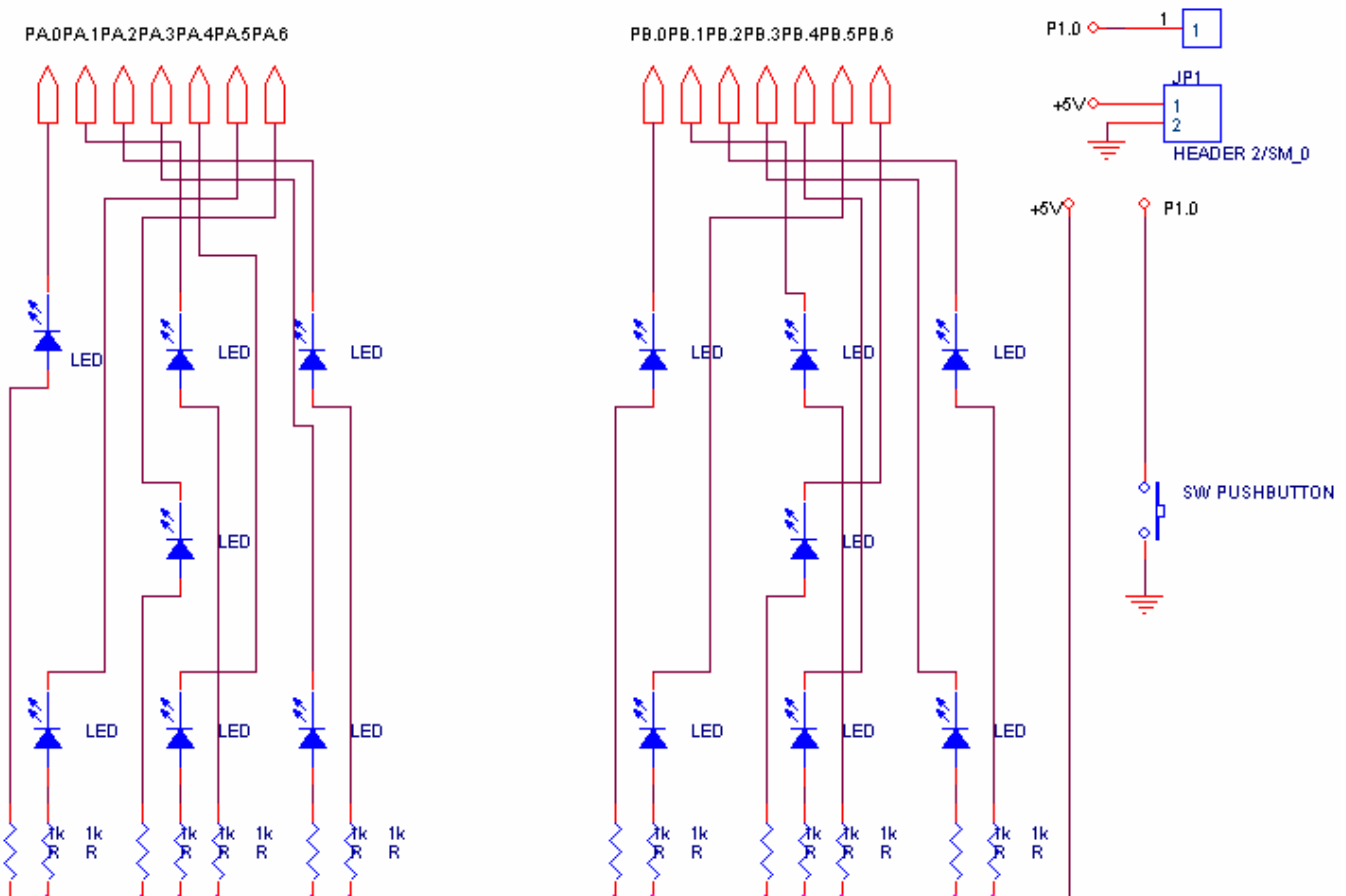
Aplikasi ini menggunakan:

- 1 modul DT-51 MinSys Ver 3.0
- 14 buah LED
- 14 buah resistor 1K Ω
- 1 buah *tactile switch* (*push-button*)

Adapun blok diagram sistem dan hubungan DT-51 MinSys dengan rangkaian terdapat pada gambar 1 dan 2.



Gambar 1
Blok Diagram AN35

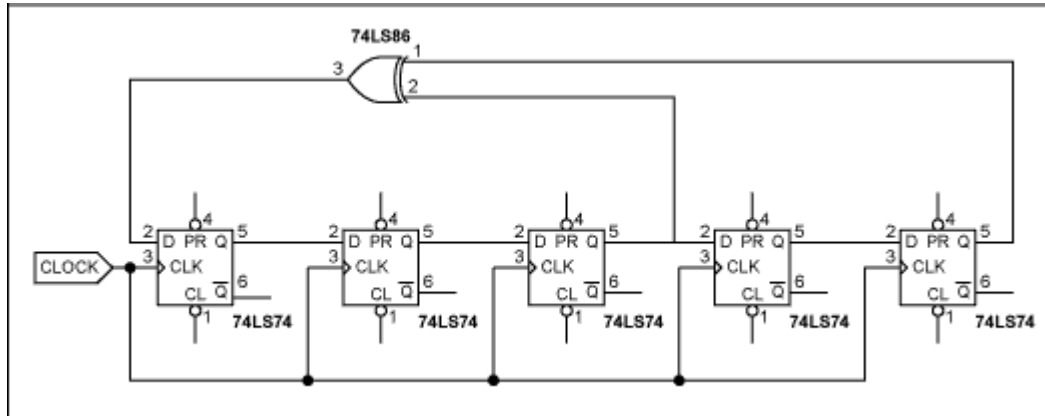


Gambar 2
Rangkaian Dadu Elektronik

Setelah semua rangkaian terhubung dan mendapat tegangan catu yang sesuai, *download*-lah Dadu.HEX ke DT-51 MinSys ver 3.0.

Konsep *Pseudo-Random Number Generator* yang digunakan ada dua jenis:

1. LFSR (*Linear Feedback Shift Register*) yang berfungsi sebagai *shift register* dengan *feedback* berupa XOR.



Gambar 3
Prinsip LFSR secara Rangkaian

Jika semua tingkatan dari flip-flop tersebut menghasilkan '0', maka *shift register* tersebut akan menjadi macet (*stuck*). Hal ini dikarenakan XOR dari '0' adalah tetap '0'. Untuk menghindari hal tersebut, maka rutin pertama kali harus diisi dengan bilangan acak (tersebar) yang bukan nol sehingga akan menghasilkan bilangan random asalkan bukan nol. Semakin panjang tingkatan dalam LFSR tersebut maka perulangan dalam *random* semakin lama.

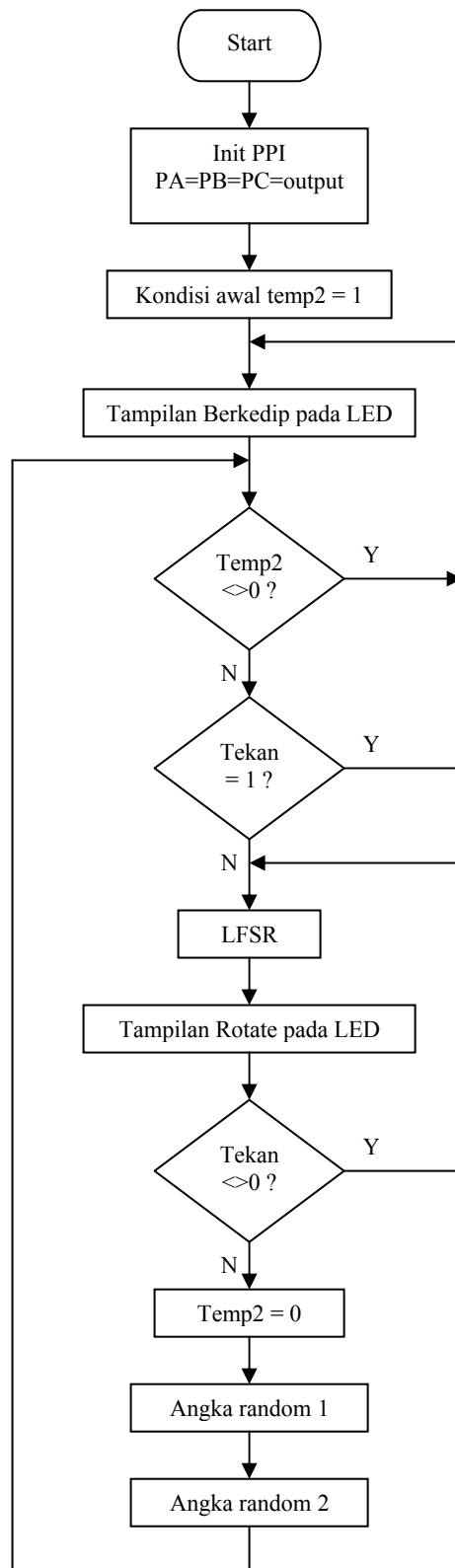
2. *Multiplicative Congruential Algorithm*. Algoritma tersebut ditunjukkan dengan menggunakan rumus sebagai berikut:

$$n_i = (a \times n_{i-1}) \text{ mod } m$$

dimana a sebagai sebuah pengali, m sebagai pembagi dan n adalah angka yang dihasilkan.

Dengan semakin besarnya angka a maka semakin banyak kemungkinan yang akan dihasilkan dan yang akan diambil adalah sisa hasil yang akan dibagi dengan m .

Flowchart dari sistem ini adalah sebagai berikut:



Gambar 4
Flowchart Program

Penjelasan

1. Inisialisasi terhadap PPI dengan ketiga *port* sebagai *output*.
2. LED ditampilkan berkedip pada kedua “dadu” secara bergantian dengan program berikut.

```
Awal:      MOV    A, #0FFH
           CPL    A
           MOV    DPTR, #PortA
           MOVX   @DPTR, A
           CPL    A
           MOV    DPTR, #PortB
           MOVX   @DPTR, A
           CPL    A
```

Pada awalnya *accumulator* diisi dengan FFh kemudian dikomplemenkan dan ditampilkan pada portA kemudian dikomplemenkan lagi dan ditampilkan pada portB lalu dikomplemenkan lagi sehingga tampilan pada portA dan portB nyala dan padam secara bergantian.

3. Temp2 diperiksa apakah nilainya sama dengan nol. Jika “ya” maka tampilan kembali berkedip. Jika “tidak”, maka tombol pada P1.0 diperiksa apakah ada penekanan. Jika tombol tersebut tidak ditekan, maka tampilan LED kembali berkedip. Jika tombol tersebut ditekan, maka nilai “dadu” dari nilai 1 hingga 6 akan diacak dengan LFSR yang dilakukan dengan rutin sebagai berikut:

```
Random:    MOV    A, Bag4
           RRC    A
           MOV    Bag4, Bag3
           MOV    Bag3, Bag2
           MOV    Bag2, Bag1
           MOV    A, Bag4
           RRC    A
           MOV    Tampung, A
           MOV    A, Bag1
           RLC    A
           MOV    Bag1, A
           MOV    A, Tampung
           XRL    A, Bag1
           MOV    Tampung, A
           MOV    A, Bag4
           RRC    A
           MOV    Bag4, A
           MOV    A, Bag3
           RRC    A
           MOV    Bag3, A
           MOV    A, Bag2
           RRC    A
           MOV    Bag2, A
           MOV    A, Tampung
           RLC    A
           MOV    Bag1, A
           MOV    Angka1, Bag1
           MOV    Angka2, Bag3
```

Variable Bag1 harus diisi dengan bukan angka nol.

Dari LFSR ini digunakan 25 tingkatan (25 bit) yang setiap variable terdiri dari 8 bit (bag1, bag2, bag3, bag4) dan *bit* ke-25 adalah carry. Dari potongan program tersebut dilakukan penggeseran seperti halnya *shift register* dan *tap*-nya adalah *bit* ke-7 dan *bit* ke-25 yang kemudian di-XOR dan hasilnya diletakkan pada *bit* 1. Bag1 diambil sebagai angka *random* pada dadu 1 dan bag 3 diambil sebagai angka *random* dadu 2.

4. Setelah selesai melakukan LFSR maka ditampilkan *Rotate* LED yaitu seperti lampu berjalan.
5. Tombol diperiksa lagi apakah telah dilepas (tidak ditekan) atau tidak. Jika “tidak” maka dilakukan LFSR lagi, jika “ya” maka lanjut ke eksekusi selanjutnya.
6. Temp2 diisi dengan 0H agar setelah terjadi penekanan tombol tidak akan ditampilkan LED berkedip lagi pada portA dan portB.

7. Langkah selanjutnya adalah *Multiplicative Congruential Algorithm* dengan rutin sebagai berikut:

```
Bagi:      MUL   AB
           MOV   R4, A
           MOV   A, B
           MOV   B, #06H
           DIV   AB
           MOV   Temp1, B
           MOV   R0, #Temp1
           MOV   A, R4
           XCHD  A, @R0
           SWAP  A
           MOV   B, #06H
           DIV   AB
           MOV   A, B
           SWAP  A
           XRL   A, @R0
           MOV   B, #06H
           DIV   AB
           MOV   R6, B
```

Dimana *accumulator* sebagai angka yang dihasilkan dari LFSR dan B sebagai pengalinya. Setelah dikalikan, maka dilakukan pembagian dengan memakai metode yang biasa dipakai:

$$\begin{array}{r} \text{Hasil} \\ \text{Pembagi} \overline{) \text{Dibagi}} \\ \hline \text{Sisa} \end{array}$$

Dari pembagian tersebut diambil sisanya yang disimpan pada R6 dan kemudian ditampilkan pada LED.

8. Setelah dihasilkan random pada dadu 1 dan dadu 2, kembali lagi dilakukan pembacaan tombol dan diulang secara terus-menerus. Jadi jika tombol ditekan maka akan menampilkan *Rotate* LED dan jika dilepas maka akan menampilkan angka acak ke kedua "dadu".

Listing program terdapat pada **AN35.ZIP**.

Selamat berinovasi!