TOYCATHON

TURBOFUTURE

(Smart Dice)

Display Code

```
byte ONE[] = {
B00010000,
B00011000,
B00010100,
B00010000,
B00010000,
B00010000,
B01111110,
B00000000};

byte TWO[] = {
B00011100,
B00100010,
B00100000,
B00010000,
B00001000,
B00000100,
B01111110,
B00000000};
```

```
byte FOUR[] = {
B00010000,
B00011000,
B00010100,
B00010010,
B01111111,
B00010000,
B00010000,
B00000000};


byte THREE[] = {
B01111100,
B00100000,
B00010000,
B00111000,
B01000000,
B01000100,
B00111000,
B00000000};


byte FIVE[] = {
B01111100,
```

```
B00000100,
B00000100,
B00111100,
B01000000,
B01000000,
B00111100,
B0000000};

byte SIX[] = {
B00010000,
B00001000,
B00000100,
B00011110,
B00100010,
B00100010,
B00011100,
B00000000};

byte random1[] = {
B01010101,
B10101010,
B01010101,
B10101010,
B01010101,
B10101010,
B01010101,
```

```
B10101010};

byte random2[] = {
B10101010,
B01010101,
B10101010,
B01010101,
B10101010,
B01010101,
B10101010,
B01010101};

#include <SPI.h>
const int columnPins[] = {6, 12, 11, 3, 16, 4, 8, 10};
//const int rowPins[] = {2, 7, 18, 5, 14, 17,13, 15};
void setup() {

 Serial.begin(9600);
  for (int i = 0; i < 8; i++)
  {
    pinMode(rowPins[i], OUTPUT);
    pinMode(columnPins[i], OUTPUT);
    digitalWrite(columnPins[i], LOW);
  }
}
```

```
void loop()
{

  int Delay = 500;
    show (ONE, 5000);
    show (TWO, 5000);
    show (THREE, 5000);
    show (FOUR, 5000);
    show (FIVE, 5000);
   show (SIX, 5000);

  delay(Delay);
}



void randomeffect()
{
  for(int a=0;a<2;a++)
  {
    show(random1,50);
    delay(40);
    show(random2,50);
    delay(40);
  }
}
void show( byte * image, unsigned long duration)
```

```
{
  unsigned long start = millis();
  while (start + duration > millis())
  {
    for(int row = 0; row < 8; row++)
    {
      digitalWrite(rowPins[row], HIGH);

      for(int column = 0; column < 8; column++)
      {
        6oolean pixel = bitRead(image[row],column);
        if(pixel == 1)
        {
          digitalWrite(columnPins[column], LOW);
        }
        delayMicroseconds(300);
        digitalWrite(columnPins[column], HIGH);
       }
      digitalWrite(rowPins[row], LOW);
     }
  }
}
```