TOYCATHON

TURBOFUTURE

(Smart Dice)

Source Code

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

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

byte FOUR[] = {
```

```
B00100000,
B00110000,
B00101000,
B00100100,
B11111110,
B00100000,
B00100000,
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 "talkie.h"
#include <SPI.h>

Talkie voice;

const int8_t spONE[]       PROGMEM =
{0xCC,0x67,0x75,0x42,0x59,0x5D,0x3A,0x4F,0x9D,0x36,0x63,0xB7,0x59,0x
DC,0x30,0x5B,0x5C,0x23,0x61,0xF3,0xE2,0x1C,0xF1,0xF0,0x98,0xC3,0x4B,
0x7D,0x39,0xCA,0x1D,0x2C,0x2F,0xB7,0x15,0xEF,0x70,0x79,0xBC,0xD2,0x
46,0x7C,0x52,0xE5,0xF1,0x4A,0x6A,0xB3,0x71,0x47,0xC3,0x2D,0x39,0x34,
0x4B,0x23,0x35,0xB7,0x7A,0x55,0x33,0x8F,0x59,0xDC,0xA2,0x44,0xB5,0x
BC,0x66,0x72,0x8B,0x64,0xF5,0xF6,0x98,0xC1,0x4D,0x42,0xD4,0x27,0x62,0
x38,0x2F,0x4A,0xB6,0x9C,0x88,0x68,0xBC,0xA6,0x95,0xF8,0x5C,0xA1,0x0
9,0x86,0x77,0x91,0x11,0x5B,0xFF,0x0F};
const int8_t spTWO[]       PROGMEM =
{0x0E,0x38,0x6E,0x25,0x00,0xA3,0x0D,0x3A,0xA0,0x37,0xC5,0xA0,0x05,0x
9E,0x56,0x35,0x86,0xAA,0x5E,0x8C,0xA4,0x82,0xB2,0xD7,0x74,0x31,0x22,
0x69,0xAD,0x1C,0xD3,0xC1,0xD0,0xFA,0x28,0x2B,0x2D,0x47,0xC3,0x1B,0
xC2,0xC4,0xAE,0xC6,0xCD,0x9C,0x48,0x53,0x9A,0xFF,0x0F};
const int8_t spTHREE[]     PROGMEM =
{0x02,0xD8,0x2E,0x9C,0x01,0xDB,0xA6,0x33,0x60,0xFB,0x30,0x01,0xEC,0
x20,0x12,0x8C,0xE4,0xD8,0xCA,0x32,0x96,0x73,0x63,0x41,0x39,0x89,0x98,
0xC1,0x4D,0x0D,0xED,0xB0,0x2A,0x05,0x37,0x0F,0xB4,0xA5,0xAE,0x5C,0
```

```
xDC,0x36,0xD0,0x83,0x2F,0x4A,0x71,0x7B,0x03,0xF7,0x38,0x59,0xCD,0xE
D,0x1E,0xB4,0x6B,0x14,0x35,0xB7,0x6B,0x94,0x99,0x91,0xD5,0xDC,0x26,0
x48,0x77,0x4B,0x66,0x71,0x1B,0x21,0xDB,0x2D,0x8A,0xC9,0x6D,0x88,0xF
C,0x26,0x28,0x3A,0xB7,0x21,0xF4,0x1F,0xA3,0x65,0xBC,0x02,0x38,0xBB,0
x3D,0x8E,0xF0,0x2B,0xE2,0x08,0xB7,0x34,0xFF,0x0F};

const int8_t spFOUR[]     PROGMEM =
{0x0C,0x18,0xB6,0x9A,0x01,0xC3,0x75,0x09,0x60,0xD8,0x0E,0x09,0x30,0x
A0,0x9B,0xB6,0xA0,0xBB,0xB0,0xAA,0x16,0x4E,0x82,0xEB,0xEA,0xA9,0x
FA,0x59,0x49,0x9E,0x59,0x23,0x9A,0x27,0x3B,0x78,0x66,0xAE,0x4A,0x9C,
0x9C,0xE0,0x99,0xD3,0x2A,0xBD,0x72,0x92,0xEF,0xE6,0x88,0xE4,0x45,0x
4D,0x7E,0x98,0x2D,0x62,0x67,0x37,0xF9,0xA1,0x37,0xA7,0x6C,0x94,0xE4,
0xC7,0x1E,0xDC,0x3C,0xA5,0x83,0x1F,0x8B,0xEB,0x52,0x0E,0x0E,0x7E,0x
2E,0x4E,0xC7,0x31,0xD2,0x79,0xA5,0x3A,0x0D,0xD9,0xC4,0xFF,0x07};

const int8_t spFIVE[]     PROGMEM =
{0x02,0xE8,0x3E,0x8C,0x01,0xDD,0x65,0x08,0x60,0x98,0x4C,0x06,0x34,0x9
3,0xCE,0x80,0xE6,0xDA,0x9A,0x14,0x6B,0xAA,0x47,0xD1,0x5E,0x56,0xAA
,0x6D,0x56,0xCD,0x78,0xD9,0xA9,0x1C,0x67,0x05,0x83,0xE1,0xA4,0xBA,0
x38,0xEE,0x16,0x86,0x9B,0xFA,0x60,0x87,0x5B,0x18,0x6E,0xEE,0x8B,0x1
D,0x6E,0x61,0xB9,0x69,0x36,0x65,0xBA,0x8D,0xE5,0xE5,0x3E,0x1C,0xE9,0
x0E,0x96,0x9B,0x5B,0xAB,0x95,0x2B,0x58,0x6E,0xCE,0xE5,0x3A,0x6A,0x
F3,0xB8,0x35,0x84,0x7B,0x05,0xA3,0xE3,0x36,0xEF,0x92,0x19,0xB4,0x86,0
xDB,0xB4,0x69,0xB4,0xD1,0x2A,0x4E,0x65,0x9A,0x99,0xCE,0x28,0xD9,0x
85,0x71,0x4C,0x18,0x6D,0x67,0x47,0xC6,0x5E,0x53,0x4A,0x9C,0xB5,0xE2,
0x85,0x45,0x26,0xFE,0x7F};

const int8_t spSIX[]      PROGMEM =
{0x0E,0xD8,0xAE,0xDD,0x03,0x0E,0x38,0xA6,0xD2,0x01,0xD3,0xB4,0x2C,
0xAD,0x6A,0x35,0x9D,0xB1,0x7D,0xDC,0xEE,0xC4,0x65,0xD7,0xF1,0x72,0
x47,0x24,0xB3,0x19,0xD9,0xD9,0x05,0x70,0x40,0x49,0xEA,0x02,0x98,0xBE,
0x42,0x01,0xDF,0xA4,0x69,0x40,0x00,0xDF,0x95,0xFC,0x3F};


const int speaker = 9;

const int columnPins[] = {6, 11, 10, 3, A3, 4, 8, 0};

const int rowPins[] = {2, 7, A5, 5, 13, A4, 12, 1};
```

```
void setup() {
  pinMode(speaker, OUTPUT);


  Serial.begin(9600);
  for (int i = 0; i < 8; i++)
  {
    pinMode(rowPins[i], OUTPUT); // make all the LED pins outputs
    pinMode(columnPins[i], OUTPUT);
    digitalWrite(columnPins[i], LOW); // disconnect column pins from Ground
  }
}

void loop()
{

  int Delay = 500 ;
  int x = analogRead(A0);
  int y = analogRead(A1);
  int z = analogRead(A2);//create random number between 1 and 6
  int num=random(1,32767);

  num = num%6+1;



  if(x>=329 && x<=333 && y>=329 && y<=334 && z>=399 &&
z<=404||x>=396 && x<= 400 && y>= 330 && y<=335 && z>= 333 &&
z<=338||x>=331 && x<=335 && y>=327 && y<=331 && z>=265 &&
z<=270|| x>=263 && x<=266 && y>=328 && y<=332 && z>=334 &&
z<=339||x>=326 && x<=330 && y>=260 && y<=264 && z>=331 &&
z<=334||x>=331 && x<=334 && y>=396 && y<=398 && z>=334 &&
z<=337)
  {
   if(num == 1) {
    randomeffect();
    voice.say(spONE);
    show(ONE, 5000);
   }

  else if(num == 2)
  {
    randomeffect();
    voice.say(spTWO);
    show(TWO, 5000);
  }
  else if(num == 3){
    randomeffect();
    voice.say(spTHREE);
    show(THREE, 5000);
  }
  else if(num == 4){
    randomeffect();
    voice.say(spFOUR);
    show(FOUR, 5000);
  }
  else if(num == 5) {
    randomeffect();
    voice.say(spFIVE);
    show(FIVE, 5000);
  }
  else if(num == 6){
    randomeffect();
    voice.say(spSIX);
    show(SIX, 5000);
  }

  delay(Delay);
}
}


void randomeffect()
{
  for(int a=0;a<2;a++)
  {
    show(random1,50);
    delay(40);
    show(random2,50);
    delay(40);
    show(random1,50);
    delay(40);
    show(random2,50);
    delay(40);
  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);// connect row to +5 volts
   for(int column = 0; column < 8; column++)
   {
    boolean pixel = bitRead(image[row],column);
    if(pixel == 1)
    {
     digitalWrite(columnPins[column], LOW);// connect column to Gnd
    }
    delayMicroseconds(300); // a small delay for each LED
    digitalWrite(columnPins[column], HIGH); // disconnect column from Gnd

    }
```

```
      digitalWrite(rowPins[row], LOW); // disconnect LEDs
     }
   }
 }
```