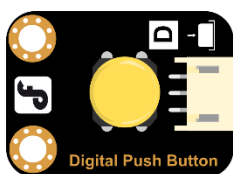


## 项目十三 数字骰子

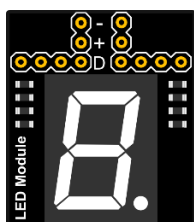
小时候都玩过飞行棋吧，是不是特别喜欢掷骰子，然而我们今天就要通过 Arduino 来做数字骰子，或者叫做电子骰子，随便怎么叫，就是这么个意思！

### 所需元件

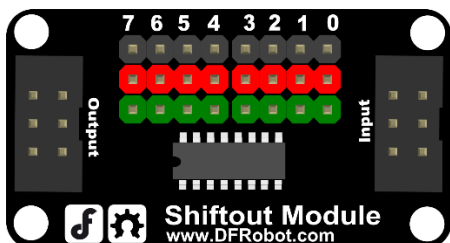
- 1× 数字大按钮模块



- 1× LED 模块

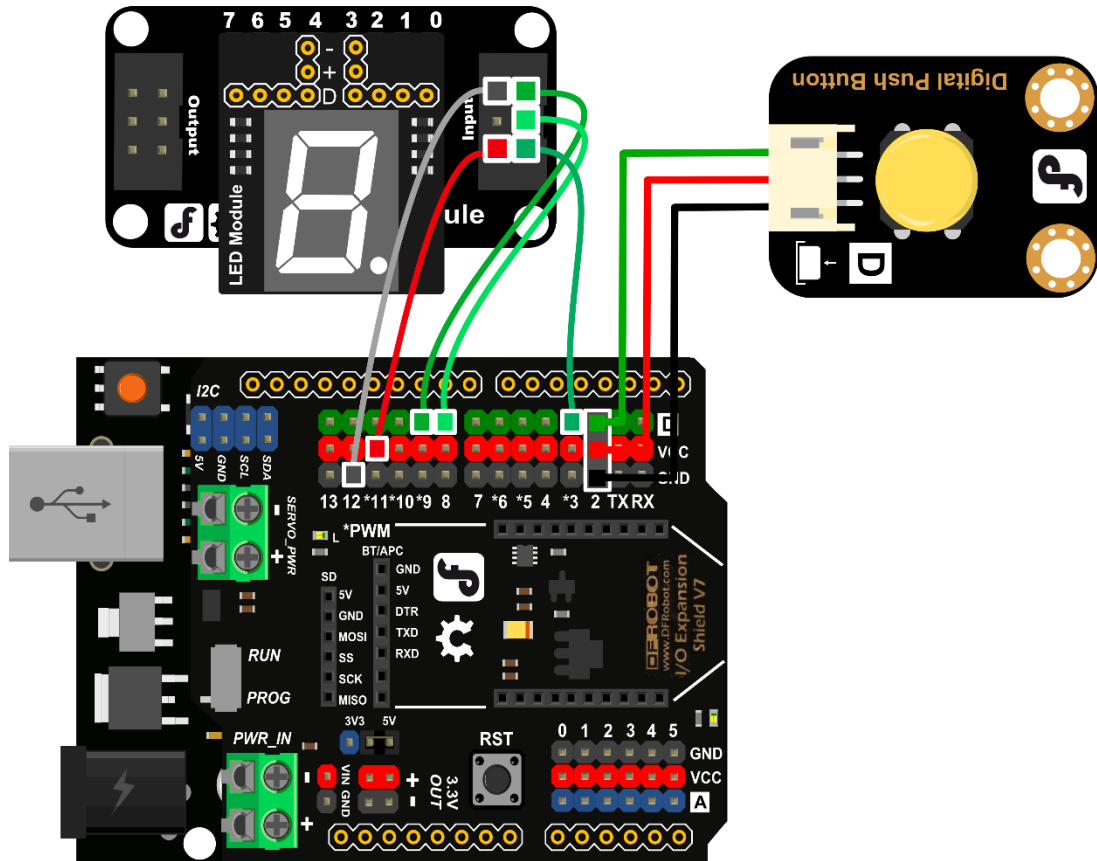


- 1× Shiftout 模块



## 硬件连接

把 LED 模块插到 Shiftout 模块上，插的时候注意引脚一一对应。LED 模块中“D”所在排连接到 Shiftout 的绿色引脚上，“+”对应红色 VCC，“-”对应黑色 GND。Input 相应接到 UNO 的数字口。数字大按钮模块接到数字口 2。



## 输入代码

样例代码 13-1:

//项目十三 - 数字骰子

```
int latchPin = 8; //数字口 8 连接到 74HC595 芯片的使能引脚
int clockPin = 3; //数字口 3 连接到 74HC595 芯片的时钟引脚
int dataPin = 9; //数字口 9 连接到 74HC595 芯片的数据引脚
int buttonPin = 2; // 按钮连接到数字口 2
```

//代表数字 0~9

```
byte Tab[]={
0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};
int number;
long randNumber;
```

```

void setup() {
    pinMode(latchPin, OUTPUT);
    pinMode(dataPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
    randomSeed(analogRead(0)); //设置一个随机数产生源模拟口 0
}

void loop(){
    randNumber = random(10); //产生 0~9 之间的随机数
    showNumber(randNumber); //显示该随机数

    //一旦有按键按下，显示该数，并保持到松开为止
    while(digitalRead(buttonPin) == HIGH){
        delay(100);
    }
}

//该函数用于数码管显示
void showNumber(int number){
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, MSBFIRST, Tab[number]);
    digitalWrite(latchPin, HIGH);
    delay(80);
}

```

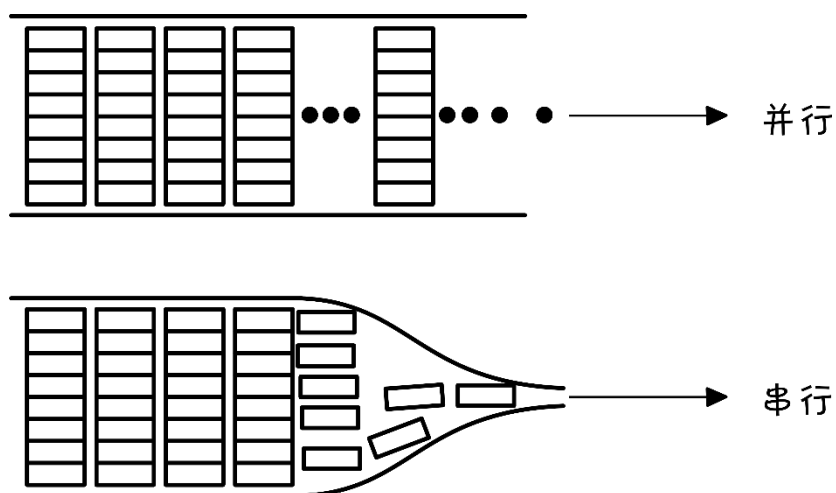
数码管会随机产生 0~9 之间的数，每次按下按钮都会是不同的数。

## 硬件分析

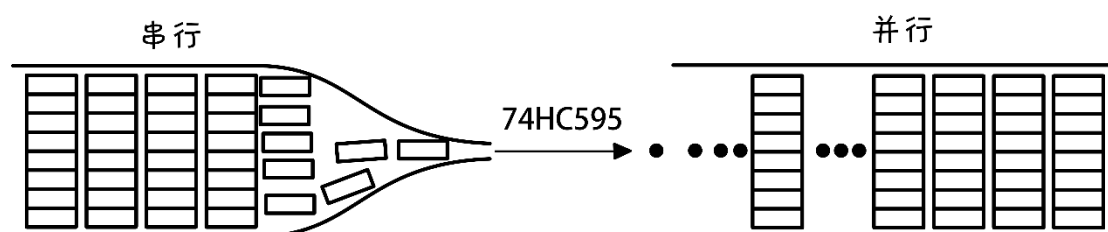
### Shiftout 模块

Shiftout 模块就是一块 74HC595 芯片。如果要看懂代码，那就需要对 74HC595 芯片的工作原理有个简单认识。我们很快的看下这块芯片是如何工作的？74HC595 实现了串行输入转并行输出的功能。

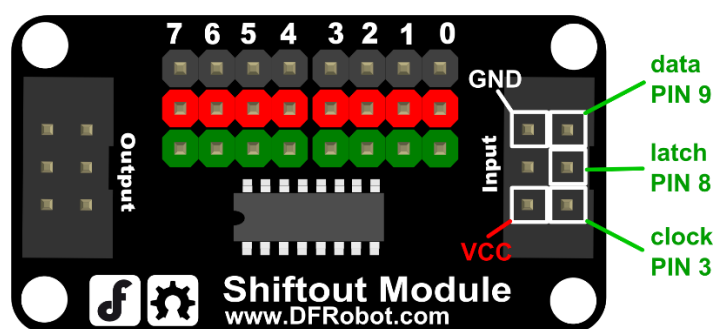
先说下什么是串行与并行。下图可以简单看出串行与并行的区别。串行，是一个一个数往发，而并行是 8 位数一道往外发的。



74HC595 可以串行进来的数据，让它并行输出。这样的好处是，比如在我们要用到多个 LED，而数字引脚又不够用的时候，用个 74HC595，就可以同时控制多个 LED 了。



那具体如何发数据，发什么数据？就是由 data, latch, clock 这三个量决定的。Arduino 提供了一个 shiftOut() 函数，使 74HC595 使用起来非常简便。



## LED 模块

LED 模块其实就是 8 个 LED，每一段都是一个独立的 LED，一共是 8 段。一个 74HC595 芯片输出正好也是 8 位，所以可以用 74HC595 的输出正好可以控制一个 LED 模块。



## 代码回顾

前面硬件介绍部分提到了 74HC595 的用法，起到的作用就是能够通过一个数据口并行输出 8 位，不会让 LED 占用 8 个数字引脚，当然如果你想接 8 个数字口也是没有问题的，只是占用的引脚会多一点而已。

也说了三个比较关键的引脚 latchPin, clockPin, dataPin。所以代码开始定义了这三个量，以及按钮。

下面就来说下 shiftOut()函数怎样用？

shiftOut 函数格式：

**shiftOut(dataPin, clockPin, bitOrder, value)**

dataPin: 输出每一位数据的引脚(int)

clockPin: 时钟引脚，当 dataPin 有值时此引脚电平变化(int)

bitOrder: 输出位的顺序，最高位优先(MSBFIRST)或最低位优先(LSBFIRST)

value: 要移位输出的数据(byte)

注意：(1) dataPin 和 clockPin 要 setup()的 pinMode()中，设置为 OUTPUT。

(2) shiftOut 目前只能输出 1 个字节(8 位)，所以如果输出值大于 255 需要分两步。

相关参考资料：

[http://wiki.dfrobot.com.cn/index.php/ShiftOut\(\)](http://wiki.dfrobot.com.cn/index.php/ShiftOut())

<http://arduino.cc/en/Reference/ShiftOut>

<http://arduino.cc/en/Tutorial/ShiftOut>

代码中，我们可以看出输出位的顺序是最高位优先的，Tab[number]就是输出的数据。

```
shiftOut(dataPin, clockPin, MSBFIRST, Tab[number]);
```

那我们看下 Tab[number]里面是些什么？

```
byte Tab[]={  
0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};
```

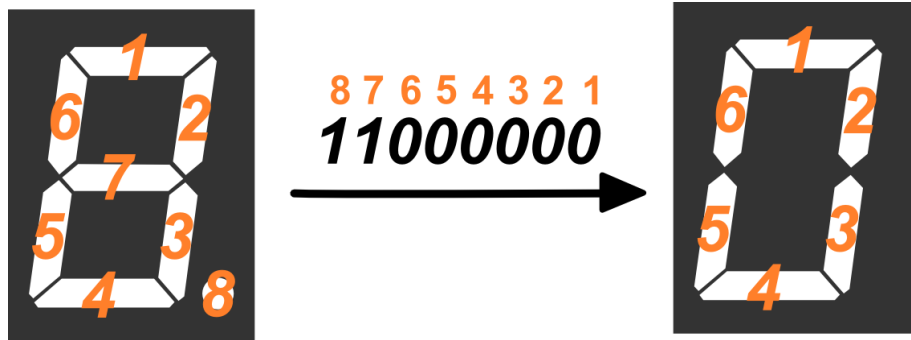
是不是看的一头雾水，什么又是 c0，又是 f9 的，这都是些什么东西？我们只要说一个，之后的就全明白了。

0xc0 这些是 16 进制表达方式，我们日常生活接触比较多的是 10 进制，也就是 0~9 的表现形式。16 进制是由 0~9，A~F 组成。0~9 对应 0~9；A~F 对应 10~15。

你可能说还是不太明白，没关系，打开电脑自带的计算器，设置成“程序员”模式。选择“十六进制”，输入“c0”，点击“二进制”。此时，“c0”就变为了“11000000”。



你会说还是没和 LED 灯对上号啊？不要急，接着往下。



看出点名堂了吗？细心的朋友应该可以看出，出来的 8 位数正好是和 LED 模块上的 8 个 LED 对应的。这里“0”为点亮，“1”为熄灭。这是由于这个共阴数码管，低电平的时候才能被点亮。这里就多做说明。其他的数字应该也能按照相同的方法推算出来。

现在我们知道了 0~9 的数字是如何显示的，既然需要做数字骰子，还有一个重要的一步，如何随机产生 0~9 之间的数字呢？Arduino 提供了个好用的函数 `random()`。

```
random (max)
```

`random ()` 可生成随机数，生成  $[0, \text{max}-1]$  范围内的随机数。`max` 是最大值。

```
random(10); // 生成 0~9 之间的数
```

`randomSeed()` 函数是用来设置随机种子的，我们这里就接到了模拟口 0。

```
randomSeed(analogRead(0));
```

## 趣味练习

LED 模块还有其他的玩儿，比如可以结合我们前面的红外接收管，做个红外遥控数码管，在数码管上显示你在红外遥控器上按下的数字。没事儿玩儿猜数字游戏也不错啊~

希望你的 Arduino 之旅不会因此而停止，用你的奇思妙想，玩出更多新颖有创意的作品。如果你愿意与我们分享的话，也可以直接登陆我们的论坛，让我们的社区论坛记录下你的点点滴滴！

欢迎登陆 DFRobot 创客社区！

DFRobotcc 创客社区：[www.dfrobot.com.cn](http://www.dfrobot.com.cn)