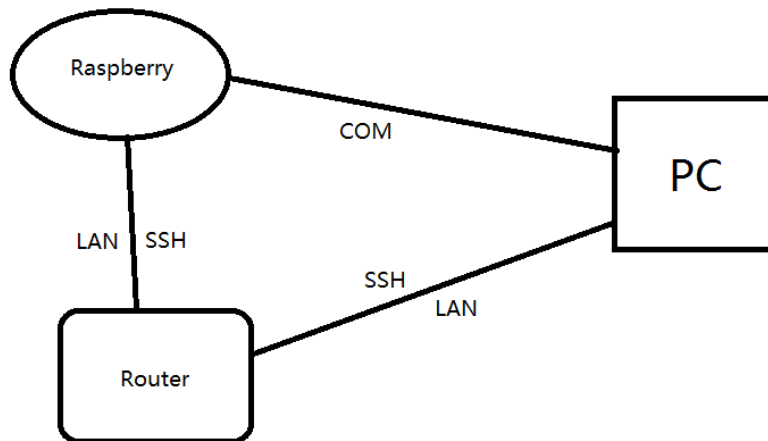


智能骰子（四位超级版 0-9999）：

HINT:已经在必做实验中实现该功能

说明，按一下开始抛骰子，再按一下，停止抛骰子。

连接示意图：



使用器材：

PC 一台（虚拟机 Ubuntu14.10+WIN8）

树莓派板子一块

TP-LINK 无线路由器

网线

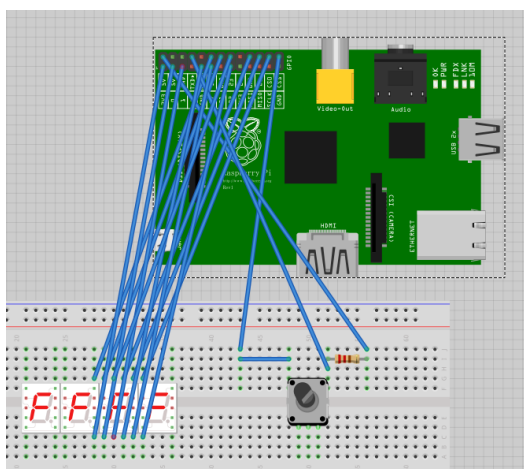
串口转 USB 线

树莓派电源线

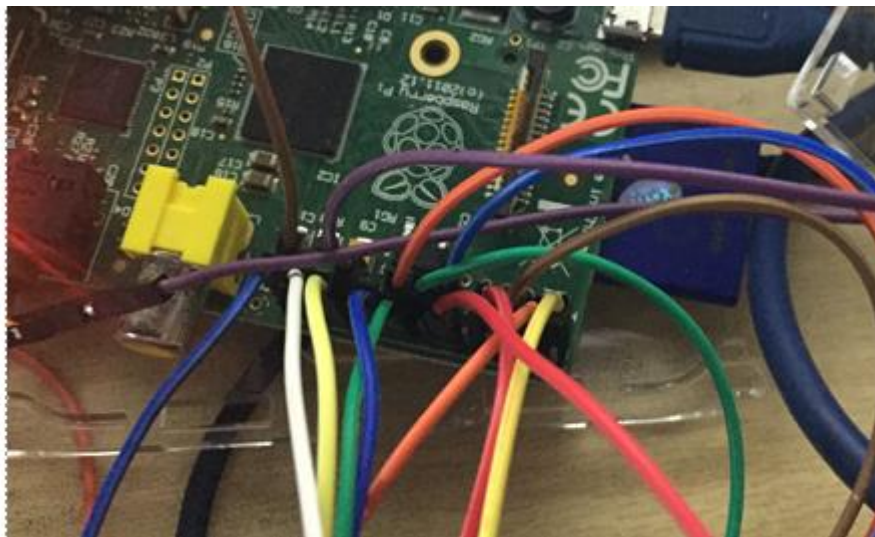
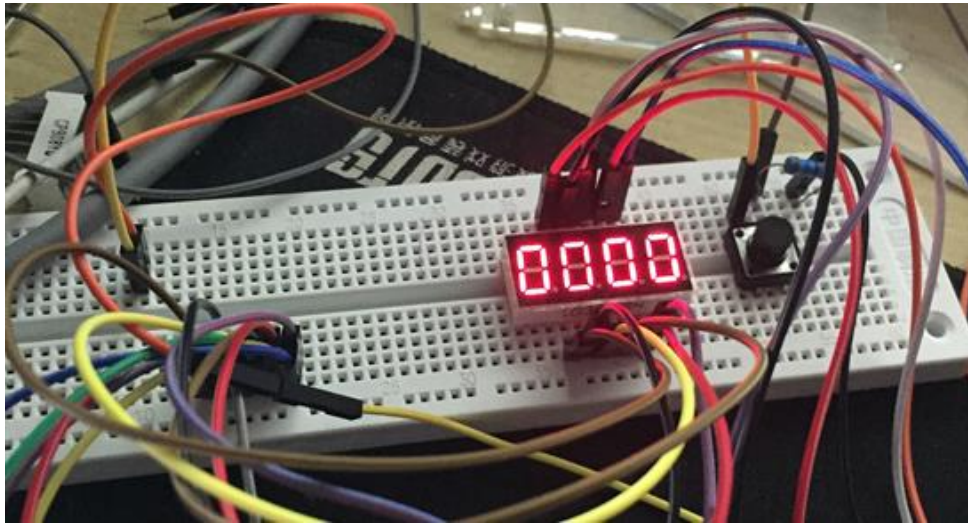
实验步骤：

安装树莓派 GPIO 口控制的相应组件：

连线：



实物连线：



根据引脚情况写代码：（单个开关，按一下开启，按一下关闭）

```
#define BUTTON 12

int begin, former, k;
int pin[12] = {1,9,7,3,2,8,6,10,11,0,4,5};
int digit[10][8] = { //a b c d e f g dp
    {0,0,0,0,0,0,1,1},
    {1,0,0,1,1,1,1,1},
    {0,0,1,0,0,1,0,1},
    {0,0,0,0,1,1,0,1},
    {1,0,0,1,1,0,0,1},
    {0,1,0,0,1,0,0,1},
    {0,1,0,0,0,0,0,1},
    {0,0,0,1,1,1,1,1},
    {0,0,0,0,0,0,0,1},
    {0,0,0,1,1,0,0,1}
};
```

开关引脚的定义，以及一些全局变量的定义，在数码管上显示数字的显示矩阵。

```

void write_pos(int pos){
    int i = 0;
    pos %= 4;
    for(; i<4; ++i){
        if(i==pos)
            digitalWrite(pin[i], HIGH);
        else
            digitalWrite(pin[i], LOW);
    }
}

```

点亮指定位置的数码管，关闭其他三个数码管。

```

void write_seg(int pos, int num){
    int i;
    num %= 10;
    write_pos(pos);
    for(i=4; i<12; i++)
        digitalWrite(pin[i], (digit[num][i-4]==0)?LOW:HIGH);
}

```

写数字

```

void segment(int show_num){
    int flag0, flag1, i;
    i = 2;
    flag0 = flag1 = 0;
    if(digitalRead(BUTTON)==1 && former==0){
        flag0=1;
    }
    else if(digitalRead(BUTTON)==0){
        flag1=1;
    }
    write_seg(3, show_num/1000);
    delay(5);
    write_seg(2, show_num/100);
    delay(5);
    write_seg(1, show_num/10);
    delay(5);
    write_seg(0, show_num);
    delay(5);
    if(digitalRead(BUTTON)==1&&flag0==1){
        begin=!begin;
        if(begin==0)
            k=1;
        former=1;
    }
    if(digitalRead(BUTTON)==0&&flag1==1)
        former=0;
}

```

扫描程序，开关的检测，former 储存开关的上一个状态，来判断这次按下是新的又一次按下还是上次依旧保持按住不变的结果。（利用扫描需要的延时来做开关的防抖动）

```

void initial(){
    int i;
    pinMode(BUTTON, INPUT);
    for(i = 0; i < 12; i++)
        pinMode(pin[i], OUTPUT);
}

```

各个引脚的初始化

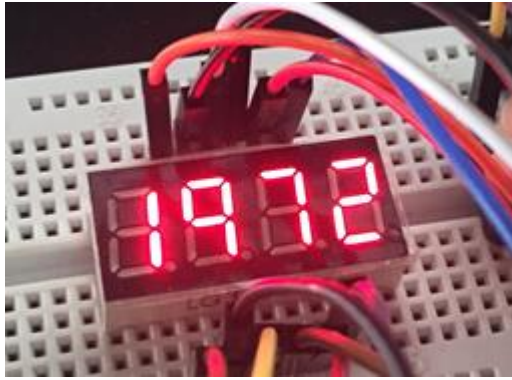
```
int main(){
    int num, i;
    if(wiringPiSetup()==-1)
        exit(1);
    srand((unsigned int)time(0));
    initial();
    begin = former = 0;
    num = k = 0;
    while(1){
        if(begin)
            num = rand()%10000;
        else if(k==1){
            printf("Result: %d\n", num);
            k=0;
        }
        //num=5234;
        segment(num);
    }
    return 0;
}
```

主函数：随机数字的生成，结果的 shell 输出。

结果运行：

实物：





Shell:

```
Result: 9104  
Result: 6942  
Result: 1972
```

短视频链接:

优酷搜索 hhhhsdxxx1993 的 task 视频

http://v.youku.com/v_show/id_XMTI1NDA5MDM1Mg==.html