

# Arduino 课题实验报告

——贪吃蛇小游戏的制作

16307110219 李进之

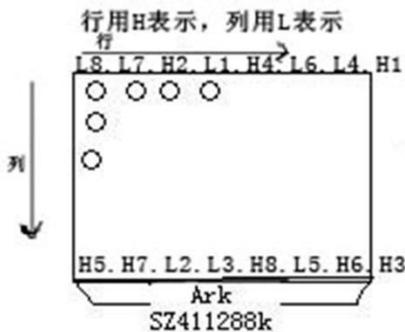
**摘要：**本实验用 Arduino 单片机制作了贪吃蛇小游戏，玩家可以通过串口输入想要移动的方向从而达到游戏的互动效果。

## 一、引言：

Arduino 是一个基于易使用的硬件和软件的开源电子平台，适用于进行交互式项目。贪吃蛇是广为人知的益智类小游戏，可训练人的敏捷性，在 Arduino 基础教程中，贪吃蛇课题常常作为毕业性的测试 project，但是往往因为硬件软件不配套，我们需要自己动手进行硬件测试、代码编写。本课题中我们用最少的器材，在 8\*8 LED 点阵上模拟实现了贪吃蛇移动捕食的动画。

## 二、实验原理：

1. 在 Arduino 软件 IDE 端进行编程，通过数据线将程序输入单片机中执行，程序逻辑控制 UNO 板 I/O 端输出高、低电压，从而控制 8\*8LED 阵点的亮灭。
2. 用户通过串口在适当时机输入“方向”，改变蛇的移动。
3. 8\*8LED 点阵阵脚分布图：



值得注意的是我们买到的 LED 可能有很多种型号，而其中每个阵脚对应哪行哪列可能需要自己测。事实上行与列都是交错排布，以上图为例，H 代表行号，L 代表列号。

8\*8LED 点亮的逻辑在于：如果该点的行阵脚为高电压、列阵脚为低电压，那么就导通，其他三种情况（H 低 L 低、H 低 L 高、H 高 L 高）都不点亮。而我们控制 LED 电亮的方式就在于：

- a) 选中某一系列阵脚，将其设为低电压，将该列需要点亮的点的行阵脚设为高电压（其余为低电压），点亮他们 1ms；
- b) 将列阵脚设为高电压，行阵脚设为低电压，使得所有点都不点亮，清空显示；
- c) 选择下一列，循环遍历所有列，重复 a、b 过程。

由于视觉暂留的效果，两个阵点亮的周期为 0.08s，我们看起来就是一直是点亮的。

### 三、实验器材：

1. Arduino UNO 板；
2. 8\*8 LED 点阵；
3. 子母线。

### 四、实验过程：

1. 首先我们需要对每个阵脚控制哪一行哪一列进行测试，主要是通过将 A 排阵脚（此处分为 A、B 是为了指代方便）输入低电位，B 排输入高电位，看看点亮了哪些阵点，从而确定对应行是高电位（即在 B 排中），对应列是低电位（A 排中）。再通过拔去一两根线、或改变某根线输入来观察哪些灯灭，从而对阵脚进行推断。

经过测试，我的 LED 点阵对应分布为：

行：{13, A3, 9, 10, 2, 8, 3, 6}；

列：{A2, 4, 5, 12, 7, 11, A4, A5}；

其中数字和字母代表的是 UNO 板上的对应 I\O 接口。

2. 其次我们编写串口交互逻辑：如果串口有输入，则每次从串口读入 1 个字节，判断其数值来改变贪吃蛇前进方向；否则维持原方向不变。

3. 编写贪吃蛇运行逻辑：

a) 判断按照指定方向移动蛇头是否将碰到墙壁，如果是，输出对应的错误，比如碰到上方墙壁则在串口输出“Error1”。

b) 判断按照指定方向移动蛇头是否将吃到食物，如果是，则将食物坐标变为蛇头坐标，与之前的蛇身接合起来。同时生成一个新的食物坐标，如果该坐标与蛇身坐标重合，则重新生成直到不再是蛇身坐标。

c) 如果该次移动未吃到食物，则将蛇身整体按照对应方向移动。

d) 每移动一次，LED8\*8 电阵点亮 150 次（大约 1.5s），用点亮次数控制蛇的移动速度。

e) 判断已经吃到的食物个数，如果已经吃到 8 个则游戏结束，串口输出“WIN”。

### 五、实验结果：

程序运行如图：

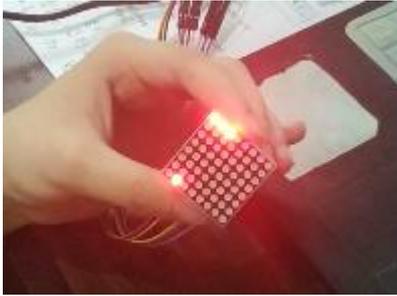
1. 蛇身仅两个点



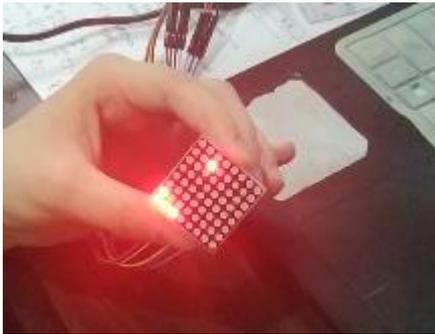
2. 蛇快要吃到食物



3. 蛇吃到食物，变为 3 个点，重新生成食物：



4. 蛇吃到食物，变为 4 个点，重新生成食物。



#### 六. 实验结论：

本次实验用 8\*8LED 点阵实现了贪吃蛇小游戏，其中最重要的是学会了控制 LED 各个点亮灭的方式（用 16 维阵脚控制 64 维点阵），以及对串口信息交互的掌握，并在学习过程中锻炼了 Arduino 编程技巧。

#### 七. 参考文献：

1. <https://blog.csdn.net/ling3ye/article/details/52986071>

#### 八. 附录：Arduino 贪吃蛇代码：

```
int R[] = {13, A3, 9, 10, 2, 8, 3, 6}; //行
int C[] = {A2, 4, 5, 12, 7, 11, A4, A5}; //列

void setup() {
```

```

Serial.begin(9600); // Start serial Serial at 9600 baud
for(int i = 0;i<8;i++)
{
    pinMode(R[i], OUTPUT);
    pinMode(C[i], OUTPUT);
}
}

int Fx,Fy; //食物坐标
int Sx=6,Sy=7; //蛇头的坐标
char s[20]={6,7,7,7}; //蛇身体初始坐标集合,一共2个点,4个元素;同时
为蛇身长度设置上限,最多吃8个食物
int s_len=4;
int dic=1; //“蛇头”前进方向,1是上,2是下,3是左,4是右

void Fgen() { //生成新的食物坐标
    while(HIGH) {
        Fx=int(random(8));
        Fy=int(random(8));
        int i=0;
        for(i; i<s_len; i+=2){
            if (s[i]==Fx && s[i+1]==Fy) { //如果食物坐标是蛇身体的一部分
                break; // 跳出 for 循环, 继续 while 循环
            }
        }
        if(i == s_len) { // 如果循环是正常结束的,那么跳出 while 循环
            break;
        }
    }
}

int turn = 0; //吃到食物个数

void loop() {
    if (turn==8) { //当吃到8个食物之后就游戏结束
        while(1) {
            Serial.println("WIN");
            delay(1000);
        }
    }
    else{
        if ( Serial.available() > 0 ) // if data is available to read

```

```

{
  int y=Serial.read();
  Serial.println(y);
  if (y=='1')
    {dic=1;}
  else if (y=='2')
    {dic=2;}
  else if (y=='3')
    {dic=3;}
  else if (y=='4')
    {dic=4;}
}

// 蛇头先运动
if (dic==1)
{ if (Sx==0){
  while(1){
    Serial.println("Error1");
    delay(1000);} //碰到上边界
  }
  else {Sx=Sx-1;} }
else if (dic==2)
{ if (Sx==7){
  while(1){
    Serial.println("Error2");
    delay(1000);} //碰到下边界
  }
  else {Sx=Sx+1;} }
else if (dic==3)
{ if (Sy==0){
  while(1){
    Serial.println("Error3"); //碰到左边界
    delay(1000);}
  }
  else {Sy=Sy-1;} }
else if (dic==4)
{ if (Sy==7){
  while(1){
    Serial.println("Error4"); //碰到右边界
    delay(1000);}
  }
  else {Sy=Sy+1;} }

if (Sx==Fx && Sy==Fy){ //吃到食物了

```

```

    turn+=1;
    s_len+=2; //蛇身加长
    for(int i=s_len-1; i>1; i-=1){ //从第最后一个点开始
        s[i]=s[i-2];
    }
    s[0]=Sx;
    s[1]=Sy;
    Fgen(); //生成新的食物位置
}
else{
    for(int i=s_len-1; i>1; i-=1){
        s[i]=s[i-2]; //丢掉之前的最后两个点,挪动蛇身
    }
    s[0]=Sx;
    s[1]=Sy;
}

unsigned char led[8][8]= {
    0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,
    0,0,0,0,0,0,0,0,
};
for(int i=0; i<s_len; i+=2){
    led[s[i]][s[i+1]]=1;
}
led[Fx][Fy]=1;
for(int i=150; i>1; i-=1){ //重复点亮一段时间
    Display(led); //显示“蛇”和“食物”的情况
}
}
}

void Display(unsigned char dat[8][8]) //显示函数
{
    for(int c = 0; c<8;c++)
    {
        digitalWrite(C[c],LOW); //选通第 c 列

        //循环

```

```
    for(int r = 0;r<8;r++)
    {
        digitalWrite(R[r], dat[r][c]);
    }
    delay(1);
    Clear(); //清空显示去除余晖
}
}
void Clear() //清空显示
{
    for(int i = 0;i<8;i++)
    {
        digitalWrite(R[i],LOW);
        digitalWrite(C[i],HIGH); // 将正负极接反,使得所有光被清除
    }
}
```