

Arduino编程之---如何让你Arduino以状态机方式运行

技术宅Roy liudaosixway 2015-11-18

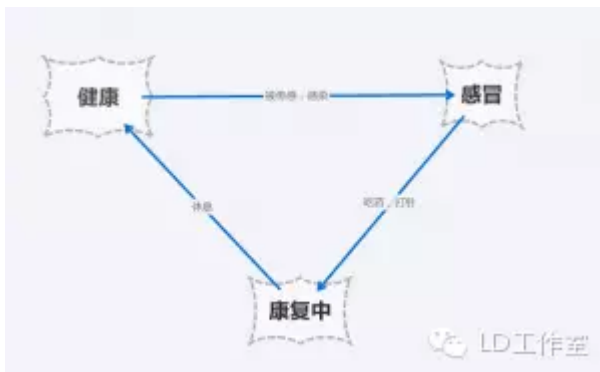


首先一个问题，何为状态机？

不用去百度状态机那些艰深的定义。简单的描述下就能知道。

在这里我所说状态机是一种编程方式，在现实中是广泛存在的。

以一个人生病为例。



健康 感冒 康复中分别就是3个状态。

状态与状态之间转换的条件就叫状态转换条件

用这种方法编程就叫有限状态机（FSM）。

其思想广泛应用于硬件控制电路设计，也是软件上常用的一种处理方法(软件上称为FMM--有限消息机)。

它把复杂的控制逻辑分解成有限个稳定状态，在每个状态上判断事件，变连续处理为离散数字处理，符合计算机的工作特点。

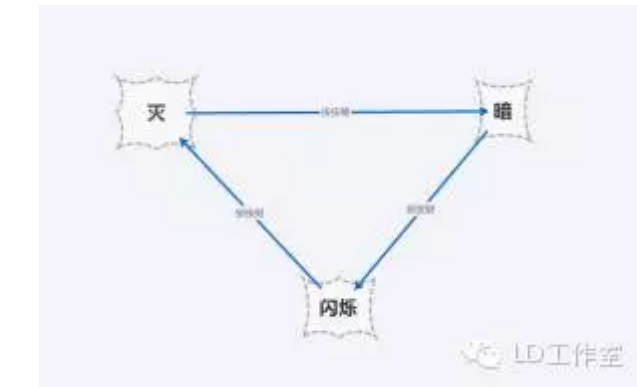
同时，因为有限状态机具有有限个状态，所以可以在实际的工程上实现。但这并不意味着其只能进行有限次的处理，相反，有限状态机是闭环系统，有限无穷，可以用有限的状态，处理无穷的事务。

重点来了，我们如何在Arduino编程中实现这种方法？

还是用实际的程序来说明吧，假设我们现在有3个状态，为简单起见现在以灯（LED）的状态来代替。

分为亮，灭，闪烁三个状态（实际应用中可以是小车的左，右，停止，再比如舵机转动，发送一个信息。。。。），状态转移条件都为按按键一次。

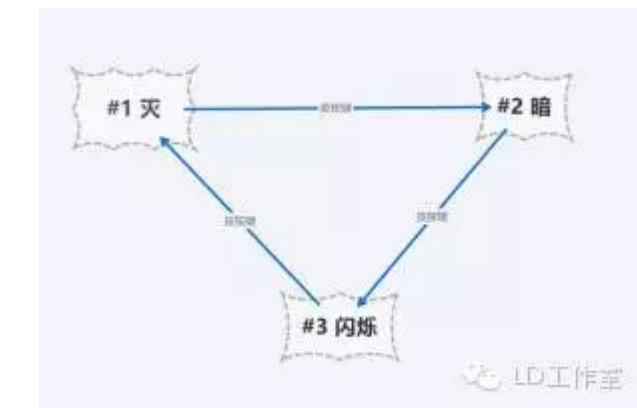
首先画出状态转移图



那么我们如何将其变为C语言呢。

一个重要的语句为Switch语句。

这个语句就是用来判断状态的，我们可以用一个标志数据来记录现在的状态并为其编号



这样我们就可以开始写代码了，下面给出参考代码。

```
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
const int led = 13;
const int buttonPin = 2;
volatile int stateFlag = 1;
// the setup routine runs once when you press reset:
void setup()
{
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
  pinMode(buttonPin, INPUT);
}
// the loop routine runs over and over again forever:
void loop()
{
  switch (stateFlag)
  {
    case 1:
      digitalWrite(led, HIGH);
      if(digitalRead(buttonPin))stateFlag=2;
      break;
    case 2:
      digitalWrite(led, LOW);
      if(digitalRead(buttonPin))stateFlag=3;
      break;
    case 3:
      LED_Blink(1000);
      if(digitalRead(buttonPin))stateFlag=1;
      break;
    default:
      break;
  }
}
void LED_Blink (unsigned int time_ms)
{
  static unsigned long int n=0;
  static int stateLed = LOW;
  if (millis()-n>=time_ms)
  {
    stateLed=!stateLed;
    digitalWrite(led, stateLed);
    n=millis();
  }
}
```

```
}  
}
```

好了，那么你的第一个状态机程序就完成了。感谢阅读

如果喜欢观看类似科技新奇事物，以及了解创客圈最新资讯，或者您对Arduino有所耳闻，可以关注我们微信公众号，一定会带给您最新的资讯，最实用的教程，以及创客最新的玩意。

微信公众号：liudaosixway

也可以加入我们Arduino技术支持qq群：329657595

