

## 使用中断(optional)

中断就是暂时停下手中的事，转而去处理一个突发事件，完成后再回到原来的位置继续做手中的事。

示例：

写作业();

有人敲门，中断发生！

执行中断函数，开门

{

走到门口();

转动门把手();

拉开门();

}

中断结束，继续写作业

/\*接着之前的位置\*/

写作业();

# 使用中断(optional)

中断的好处：不用刻意地反复检测信号，能第一时间收到信号

示例：

//如果没有中断

```
while(作业没写完)
```

```
{
```

```
    写一道题目();
```

```
    if(去门口看一眼() == 有人) 有人来了!
```

```
        开门();
```

```
}
```

//既要花不少时间去看有没有人，又有可能让别人等着

## 使用中断(optional)

语法: `attachInterrupt(digitalPinToInterrupt(引脚号), 中断函数, 模式);`

引脚号: 在Arduino UNO上2号和3号都可以

中断函数: 中断触发后要停下来去做的事

模式:

`LOW`: 当引脚电平为低

`HIGH`: 当引脚电平为高

`RISING`: 当引脚电平由低变高

`FALLING`: 当引脚电平由高变低

`CHANGE`: 当引脚电平发生变化

举例:

`attachInterrupt(digitalPinToInterrupt(2), turnOnLED(), HIGH);`

`attachInterrupt(digitalPinToInterrupt(3), recordTime(), CHANGE);`

## 使用中断(optional)

中断函数:

语法: `void interruptFunction(void)`

//中断函数没有输入输出变量, 而且中断时`delay()`会失效

举例:

```
void turnOnLED()
```

```
{
```

```
    digitalWrite(LEDPin,HIGH);
```

```
}
```

```
/*~~~~~*/
```

`volatile unsigned long` timeRecord; //中断函数只能通过访问全局变量的方法传递参数, 而且该全局变量要加`volatile`修饰

```
void recordTime()
```

```
{
```

```
    timeRecord=micros();
```

```
}
```