attachInterrupt()

说明

attachInterrupt()函数是用于为Arduino开发板设置和执行ISR（中断服务程序）用的

ISR（中断服务程序）顾名思义就是中断Arduino当前正在处理的事情而优先去执行中断服务程序。当中断服务程序完成以后，再回来继续执行刚才执行的事情。中断服务程序对监测Arduino输入有很大的用处。

我们可以使用attachInterrupt()函数，利用Arduino的引脚触发中断程序。以下列表说明支持中断的引脚有哪些：

Arduino控制板	**支持中断的引脚**
Uno, Nano, Mini	2, 3
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21
Micro, Leonardo	0, 1, 2, 3, 7
Zero	除4号引脚以外的所有数字引脚
MKR1000 Rev.1	0, 1, 4, 5, 6, 7, 8, 9, A1, A2
Due	所有数字引脚

注意

在ISR（中断服务程序）函数中，delay()函数是不工作的，而且millis()函数返回值也不再增长。在ISR（中断服务程序）运行期间Arduino开发板接收到的串口数据也可能丢失。另外ISR函数里所使用的变量应声明为volatile类型。详情请见以下”关于ISR（中断服务程序）”部分。

使用中断

中断很适合执行那些需要不断检查的工作，比如检查一个引脚上连接的按键开关是否被按下。中断更适用于很快就会消失的信号检查，比如某一个引脚用于检测脉冲信号，这个脉冲信号的持续时间可能十分短暂。如果不使用中断，那么假如Arduino开发板正在执行其它任务时，突然这个脉冲信号来了，还不等Arduino开发板完成正在执行的工作，这个脉冲信号可能就已经消失了。而使用中断，就可以确保这个转瞬即逝的脉冲信号可以很好的被Arduino开发板检测到并执行相应任务。

关于ISR（中断服务程序）

对于Arduino开发板来说，ISR（中断服务程序）是一种特殊的函数。它的特殊意味着它具有其它类型函数所不具备的限制和特点。

- ISR函数不能有任何参数。ISR也没有任何返回值。
- 通常ISR需要越短小精悍越好！另外如果您的代码中有多个ISR函数，那么每次Arduino只能运行一个ISR函数，其它ISR函数只有在当前的ISR函数执行结束以后，才能按照其优先级别顺序执行。
- millis()函数的运行依赖Arduino开发板的中断功能，因此ISR函数中的millis()函数是无法正常运行的。micros() 也是类似的情况，它只能在初始的1-2毫秒中可以运行，但是过了这1-2毫秒后就开始出现问题了。 delayMicroseconds() 不需要任何计数器就可以运行，所以delayMicroseconds() 运行是不会受到影响的。

- 一般情况下，ISR函数与主程序之间传递数据是依靠全局变量来实现的。为了确保全局变量在ISR函数中可以正常的工作，应该将可能被ISR函数中使用的全局变量声明为volatile类型。

如需更多有关中断方面的知识，请参考 Nick Gammon's notes.

语法

```
attachInterrupt(digitalPinToInterrupt(pin), ISR, mode);
```

参数

pin: 中断引脚号
ISR: 中断服务程序名
mode：中断模式

中断模式(mode)有以下几种形式：

LOW： 当引脚为低电平时触发中断服务程序
CHANGE： 当引脚电平发生变化时触发中断服务程序
RISING： 当引脚电平由低电平变为高电平时触发中断服务程序
FALLING： 当引脚电平由高电平变为低电平时触发中断服务程序

返回值

无

示例

```
const byte ledPin = 13;

//用2号引脚作为中断触发引脚
const byte interruptPin = 2;

volatile byte state = LOW;

void setup() {
  pinMode(ledPin, OUTPUT);

  //将中断触发引脚（2号引脚）设置为INPUT_PULLUP（输入上拉）模式
  pinMode(interruptPin, INPUT_PULLUP);

  //设置中断触发程序
  attachInterrupt(digitalPinToInterrupt(interruptPin), blink, CHANGE);
}

void loop() {
  digitalWrite(ledPin, state);
}

//中断服务程序
void blink() {
  state = !state;
}
```

注意

- 在中断服务程序中，不能使用delay()函数和millis()函数。因为他们无法在中断服务程序中正常工作。delayMicroseconds()可以在中断服务程序中正常工作。
- 中断服务程序应尽量保持简单短小。否则可能会影响Arduino工作。
- 中断服务程序中涉及的变量应声明为volatile类型。
- 中断服务程序不能返回任何数值。所以应尽量在中断服务程序中使用全局变量。