

**导语：**在电机控制或者开关电源软件设计中，中断几乎是必用的功能。很多情况下，如何优化中断算法开销成了算法工程师的核心工作。DSP 芯片初学者很有必要掌握中断的原理机制、配置方式。本期我们做一个简单中断程序：timer 中断控制 LED。

## 什么是中断？

在实时控制 (real-time control) 中，代码一般顺序执行或者跳转执行。当需要处理某些实时性要求较高的程序时，软件或硬件向 CPU 发出中断请求，如果请求被接受，CPU 将暂停当前指令、保存状态并执行中断服务程序 (interrupt service routine, ISR)。执行完 ISR 后，CPU 再恢复中断前的状态，并继续运行。

与跳转指令不同的是，中断是无法预测的，一旦设置使能后，只要满足触发条件，就能产生中断。



在电机或电源的数字化控制中，中断机制是最佳的算法实现手段。毫不夸张地说，几乎所有的核心算法都是中断函数实现的。昊芯 28027 支持丰富的中断机制：timer, ADC, PWM, SCI, SPI 等外设都支持中断配置。

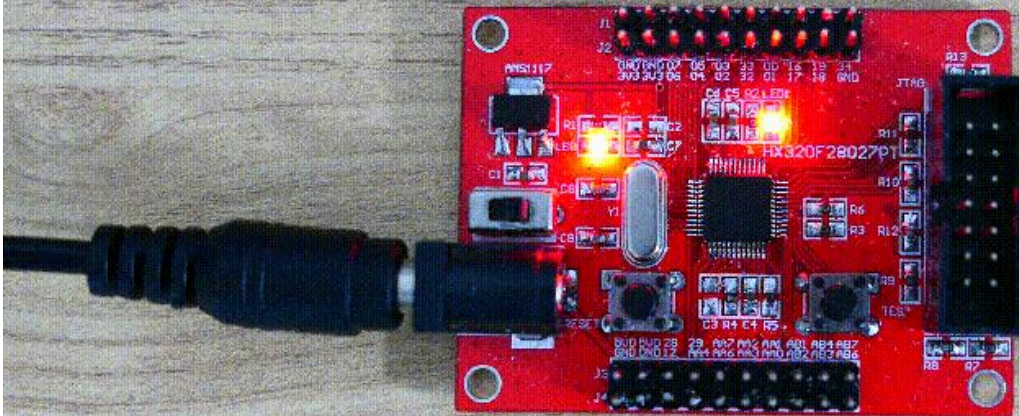
## Timer 控制 LED 的原理

昊芯 28027 内置 3 个 timer (0/1/2)。利用 Timer0 每间隔 1 秒产生一次 timer0 中断，在中断服务程序中，每次都

将 LED 对应的 GPIO 口输出电平反转，就能产生 LED 亮灭的效果。

本程序基于昊芯 28027 核心板，相关资料可以在[中科昊芯官网](#)下载。

## HXS320F28027中断演示



### 程序代码

```
/*
 * main.c
 *
 * Created on: 2021年9月25日
 * Author: daieq
 */

#include "dsc_config.h"
#include <syscalls.h>
#include "IQmathLib.h"

Uint16 IntCount = 0;

INTERRUPT void timer0_isr(void);

int main(void)
{
    //系统初始化 120MHz
    InitSysCtrl();
```

```
//GPIO 初始化
EALLOW;
GpioCtrlRegs.GPAMUX1.bit.GPIO0 = 0;
GpioCtrlRegs.GPADIR.bit.GPIO0 = 1;
GpioDataRegs.GPASET.bit.GPIO0 = 1;
EDIS;

//关闭 CPU 中断
DINT;

//初始化 PIE 模块
InitPieCtrl();

//关闭并清除 CPU 中断标志位
IER = 0x0000;
IFR = 0x0000;

//初始化 PIE 中断向量表
InitPieVectTable();

//配置中断向量表
EALLOW;
PieVectTable.TINT0 = &timer0_isr;
EDIS;

// 模块初始化
EALLOW;
CpuTimer0Regs.TCR.bit.TSS = 1;
CpuTimer0Regs.PRD.all = 120000000;
CpuTimer0Regs.TCR.bit.TIF = 1;
CpuTimer0Regs.TCR.bit.TRB = 1;
CpuTimer0Regs.TCR.bit.TIE = 1;
CpuTimer0Regs.TCR.bit.TSS = 0;
EDIS;

//使能 CPU 中断
```

```

IER |= M_INT1;

//使能 PIE 中断
PieCtrlRegs.PIEIER1.bit.INTx7 = 1;

EINT;
ERTM;

for (;;)
{
}

void CODE_SECTION("ramfuncs") INTERRUPT timer0_isr()
{
    IntCount++;

    GpioDataRegs.GPATOGGLE.bit.GPIO0 = 1;

    PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
}

```

## 关于中科昊芯

“智由芯生 创享未来”，中科昊芯是数字信号处理器专业供应商。作为中国科学院科技成果转化企业，瞄准国际前沿芯片设计技术，依托多年积累的雄厚技术实力及对产业链的理解，以开放积极的心态，基于开源指令集架构 **RISC-V**，打造多个系列数字信号处理器产品，并构建完善的处理器产品生态系统。产品具有广阔的市场前景，可广泛应用于数字信号处理、工业控制及电机驱动、数字电源、消费电子、白色家电等领域。