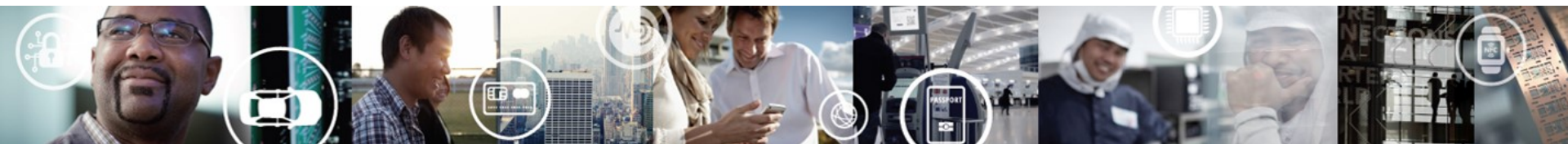


LPC82X 培训资料

外部引脚中断 动手实验

MAY, 2016



EXTERNAL USE



SECURE CONNECTIONS
FOR A SMARTER WORLD

内容

- 实验简介（目的，内容，结果）
- 软/硬件环境搭建
- 实验步骤
- 相关底层APIs说明

实验简介

- **目的：**通过本实验，理解和掌握外部引脚中断关于以下几点：
 - 配置的全过程
 - 边沿触发中断的用法
 - 电平触发中断的用法以及它与边沿触发的关键区别
- **描述：**
 - 本实验把S3键所在的PIO0_4配置为下降沿触发中断，S2键所在的PIO0_12配置为低电平触发中断，另有SysTick定时器产生周期性中断(20Hz)。
 - S3按下时在引脚中断ISR里设置软件标志——全局变量g_pinIntFlags的bit 0，并点亮LED2，待下次进入SysTick ISR时检查此标志，如果设置就熄灭LED2。
 - S2按下后在ISR里只设置软件标志——全局变量g_pinIntFlags的bit 1，待下次进入SysTick ISR时检查此标志，如果设置就切换LED3的亮和灭。
- **结果：**
 - 按下S3时LED2闪烁一次，松开时一般不闪烁（但偶尔会“抖动”闪烁）
 - 按下S2时LED3以20Hz闪烁，松开时LED3可能亮也可能灭

软/硬件环境搭建

- 硬件
 - 评估板：LPC824Lite-V1.0
- 工程位置
 - ..\peri_example\pinint\pinint_example\project_pinint_example.uvprojx

实验步骤

- 第一步 – 观看效果：编译、下载、运行，按下和放开S3和S2
- 第二步 – 调试体验：
 - 进入调试模式，单步执行PinINTConfig()函数
 - 在两个PININT ISR和SysTick ISR的条件判断内部分别设置断点，然后全速运行程序，观看断点命中的顺序。

相关底层APIs说明-1

1. (可省略)初始化PININT，只是个空函数，为保持代码可读性而保留

```
void Chip_PININT_Init(LPC_PININT_T *pPININT) {}
```

2. 配接IO引脚到PININT的某个中断源上，这是通过SYSCON模块完成的

```
void Chip_SYSCCTL_SetPinInterrupt(uint32_t intno, uint32_t pin);
```

3. (可选) 使能PININT的唤醒功能

```
void Chip_SYSCCTL_EnablePINTWakeup(uint32_t pin);
```

4. 设置引脚中断的触发方式(边沿或电平二选一)

```
void Chip_PININT_SetPinModeEdge/Level(LPC_PININT_T *pPININT, uint32_t pins);
```

相关底层APIs说明-2

5. 对于边沿触发，使能上升沿 或/和 下降沿中断；对于电平触发，选择电平

*void Chip_PININT_EnableIntHigh/Low(LPC_PININT_T *pPININT, uint32_t pins);*

*void Chip_PININT_SelectLevel(LPC_PININT_T *pPININT, uint32_t pins, bool isHigh);*

6. (仅用于边沿触发) 手动清除中断标志。电平触发由硬件自动设置和清除

*STATIC INLINE void Chip_PININT_ClearIntStatus(LPC_PININT_T *pPININT, uint32_t pins);*