

## 32 位微控制器

# HC32L15 系列的外部中断处理

### 适用对象

系列	产品型号
HC32L15	HC32L150KATA
	HC32L150JATA
	HC32L150FAUA
	HC32L156KATA
	HC32L156JATA

## 目 录

<b>1</b>	<b>摘要 .....</b>	<b>3</b>
<b>2</b>	<b>外部中断简介 .....</b>	<b>3</b>
<b>3</b>	<b>HC32L15 系列的外部中断模块 .....</b>	<b>5</b>
3.1	简介 .....	5
3.2	说明 .....	5
3.2.1	寄存器介绍.....	6
3.2.2	工作流程介绍 .....	10
<b>4</b>	<b>样例代码 .....</b>	<b>11</b>
4.1	代码介绍 .....	11
4.2	代码运行 .....	12
<b>5</b>	<b>总结 .....</b>	<b>13</b>
<b>6</b>	<b>版本信息 &amp; 联系方式 .....</b>	<b>14</b>

## 1 摘要

本篇应用笔记主要介绍 HC32L15 系列的外部中断处理，并简要说明如何使用 HC32L15 的外部中断处理配合进行外设的操作。

## 2 外部中断简介

### 什么是外部中断？

外部中断其实就是机器收到外部信号后，放下正在处理的任务，来处理你设定好的中断函数。

（引自‘百度百科’）

本文主要介绍和中断相关的控制处理部分。

相关寄存器的具体介绍读者可详见‘ARMv6-M-Architecture-Reference-Manual’以及相关的 ARM 介绍资料。

### 外部中断控制部分的操作概要

外部中断控制部分按照以下顺序将外部中断请求输出到中断控制器。

- 1) 配置外部中断电平寄存器（INTLVL），设定检测 INTxx 引脚输入信号的电平/沿。检测的电平/沿有以下 4 种选择：“H”电平，“L”电平，上升沿，下降沿。
- 2) 检测到的中断输入保留在中断向量 F/F 内，可以通过外部中断源寄存器（RIRR）读取，也可以通过外部中断源清除寄存器（INTCLR）清除。
- 3) 当外部中断通过外部中断使能寄存器（INTEN）使能时，该外部中断请求（INTIRQxx）则会被输出到中断控制器。

### 设置步骤

请按照以下步骤配置外部中断：

1. 使用外部中断使能寄存器（INTEN）禁止外部中断。
2. 使用外部中断源电平寄存器（INTLVL）设定检测条件（有效沿或者电平）。

3. 读取外部中断源电平寄存器（INTLVL）。
4. 使用外部中断源清除寄存器（INTCLR）清除外部中断源。
5. 使用外部中断使能寄存器（INTEN）使能外部中断。

## 3 HC32L15 系列的外部中断模块

### 3.1 简介

HC32L15 系列 MCU 是由华大半导体开发的基于 ARM Cortex M0+ Core 的低功耗 MCU，拥有多种定时器和多种通信外设，同时基于 MCU 的外设配合 Core 提供了较为丰富的中断源，为用户开发产品提供了更好的便利性和易用性。

### 3.2 说明

本节介绍 HC32L15 的外部中断模块，包括寄存器和工作流程。

### 3.2.1 寄存器介绍

对于外部中断模块的操作主要通过以下寄存器进行：

#### 1) 外部中断使能寄存器 (INTEN)

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	EN[15:0]															
Attribute	R/W															
Initial value	0x0000															

#### 寄存器功能描述

[bit31:16]Reserved:保留位

保留位的读取值为“0”。

[bit15:0] EN15 to EN0: 外部中断使能位

EN15~ EN0 分别对应引脚 INT15 ~ INT00。

不能设定产品规格中没有定义的引脚的对应位。

bit	描述
0	禁止对应位引脚 INTxx 的外部中断请求
1	使能对应位引脚 INTxx 的外部中断请求

寄存器中写“1”对应的中断请求会被输出到中断控制器。写“0”的对应引脚虽然保留中断源，但不输出中断请求到中断控制器。

#### 2) 外部中断源寄存器 (INTSRC)

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SRC[15:0]															
Attribute	R															
Initial value	0xFFFF															

#### 寄存器功能描述

[bit31:0]Reserved: 保留位

保留位的读取值为“0”。

[bit15:0] SRC15 to SRC0: 外部中断请求检测位

SRC15 ~ SRC0 分别对应引脚 INT15 ~ INT00。

产品规格中不存在的引脚的对应位不确定。.

bit	描述
0	未检测到对应引脚 INT <sub>xx</sub> 的外部中断请求
1	检测到对应引脚 INT <sub>xx</sub> 的外部中断请求
写	无效

### 3) 外部中断源清除寄存器 (INTCLR)

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	R/W															
Initial value	0xFFFF															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	CLR[15:0]															
Attribute	R/W															
Initial value	0xFFFF															

#### 寄存器功能描述

[bit31:16]Reserved: 保留位

保留位的读取值为“1”。

[bit15:0] CLR[15:0]: 外部中断源清除位

CLR[15:0] 对应引脚 INT15 ~ INT00.

不能将产品规格中不存在的引脚对应位写“0”。

bit	描述
写“0”	清除外部中断源
写“1”	无效
读	始终读“1”

#### 4) 外部中断电平寄存器 (INTLVL)

bit	31	30	29	28	27	26	25	24
Field	LVLH15	LVLL15	LVLH14	LVLL14	LVLH13	LVLL13	LVLH12	LVLL12
Initial value	0							
Attribute	R/W							

bit	23	22	21	20	19	18	17	16
Field	LVLH11	LVLL11	LVLH10	LVLL10	LVLH9	LVLL9	LVLH8	LVLL8
Initial value	0							
Attribute	R/W							

bit	15	14	13	12	11	10	9	8
Field	LVLH7	LVLL7	LVLH6	LVLL6	LVLH5	LVLL5	LVLH4	LVLL4
Initial value	0							
Attribute	R/W							

bit	7	6	5	4	3	2	1	0
Field	LVLH3	LVLL3	LVLH2	LVLL2	LVLH1	LVLL1	LVLH0	LVLL0
Initial value	0							
Attribute	R/W							

#### 寄存器功能描述

[bit31:0] LVLL15 ~ LVLL0 或 LVLH15 ~ LVLH0: 外部中断请求检测电平选择位 LVLL15 ~ LVLL0 和 LVLH15 ~ LVLH0 每两位组合 (LVLL 和 LVLH) 对应 INT15~INT00。

不能设定产品规格书中不存在的引脚的对应位。

一旦检测出该位选择的沿或者电平，则认为是一个外部中断请求。

LBx	LAx	描述
0	0	检测 "L" 电平
0	1	检测 "H" 电平
1	0	检测上升沿
1	1	检测下降沿

#### 5) 非屏蔽中断寄存器 (NMISRC)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															SRC
Attribute	-															R
Initial value	-															0

#### 寄存器功能描述

[bit15:1] Reserved: 保留位



读取值没有定义。写无效。

[bit0] SRC: NMI 中断请求检测位

SRC 位对应引脚 NMIX。

bit	描述
0	未检测到 NMI 中断请求
1	检测到 NMI 中断请求
写	无效

#### 6) 非屏蔽中断清除寄存器 (NMICLR)

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															CLR
Attribute	-															R/W
Initial value	-															1

#### 寄存器功能描述

[bit15:1] Reserved: 保留位

读取值没有定义。写无效。

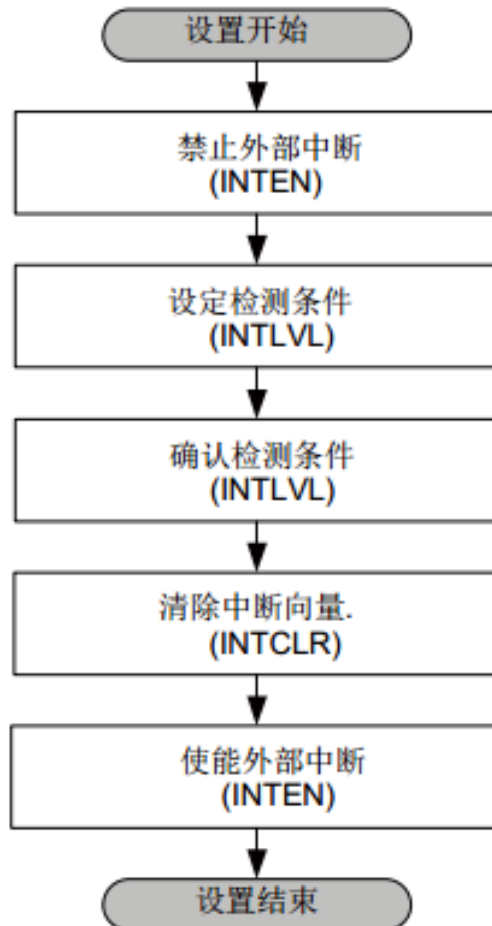
[bit0] CLR: NMI 中断清除位

CLR 位对应引脚 NMIX。

bit	描述
写 0	清除 NMI 中断请求
写 1	无效
读	读取值为“1”

### 3.2.2 工作流程介绍

外部中断的过程比较：禁止外部中断，设定检测条件，确认检测条件，清除中断向量，最后使能外部中断即可。



## 4 样例代码

### 4.1 代码介绍

用户可以根据上述的工作流程编写自己的代码来学习验证该模块，也可以直接通过华大半导体的网站下载外部中断代码（exint）。

以下部分简要介绍该代码的各个部分的功能：

- 1) 定义一个配置参数结构体：

```
stc_exint_config_t stcExintConfig;
```

- 2) 根据需要设置相应的 IO 口为外部中断口：

```
Gpio_SetFunc_INT00_0(0u);
```

- 3) 设置参数，并初始化：

```
stcExintConfig.bEnable = 1u;  
stcExintConfig.enLevel = ExIntFallingEdge;  
stcExintConfig.pfnExintCallback = Main_ExtInt0Callback;  
stcExintConfig.bTouchNvic = TRUE;  
Exint_Init(ExintInstanceIndexExint0,&stcExintConfig);
```

- 4) 等待外部中断的响应：

```
void Main_ExtInt0Callback(void)  
{  
    u32ExtInt0Count++;  
}
```

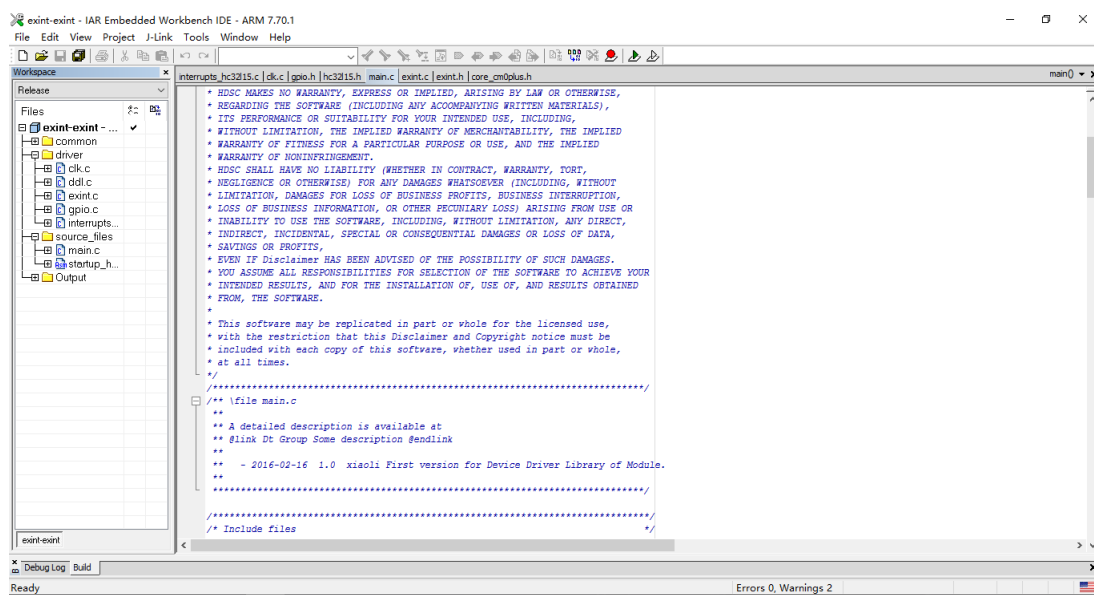
## 4.2 代码运行

用户可以通过华大半导体的网站下载到 HC32L15 的 DDL 的样例代码（exint），并配合学习板（比如‘SK-HC32L156-64L V10’）运行相关代码学习使用 exint 外设的中断操作。

以下部分主要介绍如何在‘SK-HC32L156-64L V10’学习板上运行 exint 样例代码并观察结果：

- 确认安装正确的 IAR EWARM V7.70 工具（请从 IAR 官网自行下载并安装）。
- 获取‘SK-HC32L156-64L V10’学习板。
- 从华大半导体网站下载 HC32L15 DDL 代码。
- 下载并运行 exint\exint\中的项目文件：

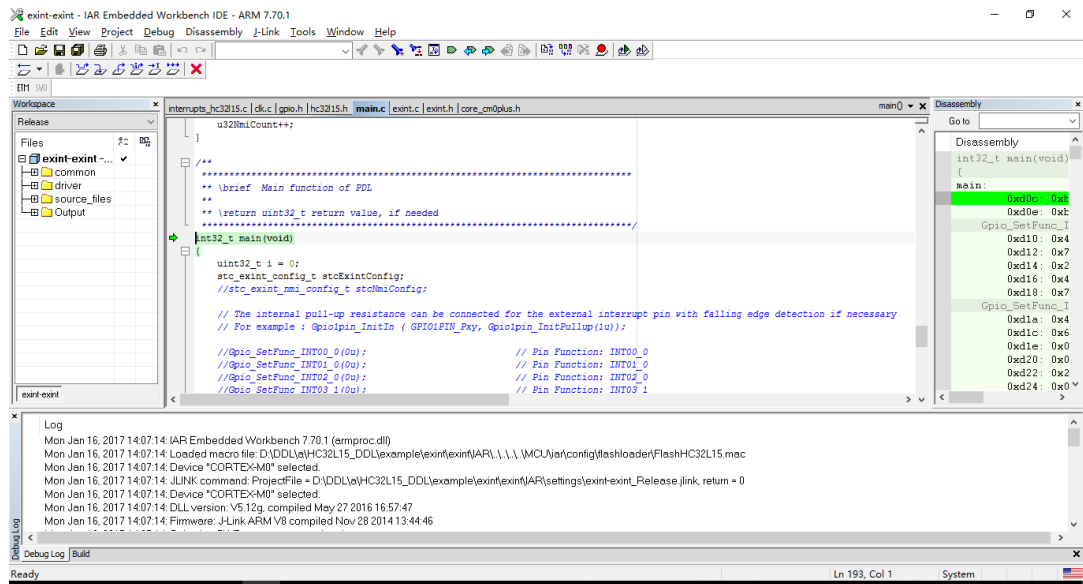
1) 打开 exint\exint\项目，并打开‘main.c’如下视图：



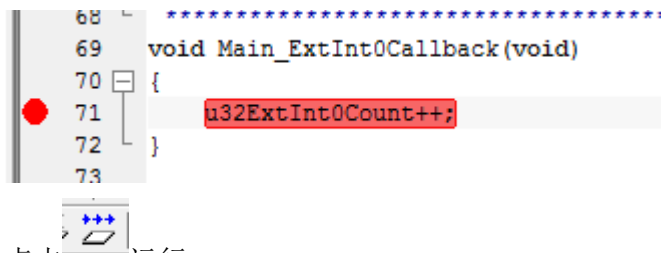
2) 点击  重新编译整个项目；

3) 点击  将代码下载到学习板上；

4) 可以看见类似如下的视图:



5) 在 ‘main.c’ 的第 71 行设置断点, 如下图所示:



6) 点击  运行。

7) 代码运行并给予相应的外部信号, 将会停止在 ‘main.c’ 的断点处, 观察并记录。

## 5 总结

以上章节简要介绍了中断处理, 详细说明了 HC32L15 系列的外部中断模块并且演示了如何使用相关的外部中断样例代码学习外部中断的设定使用, 在开发中用户可以根据自己的实际需要使用外部中断模块。

## 6 版本信息 & 联系方式

日期	版本	修改记录
2018/8/9	Rev1.0	初版发布。



---

如果您在购买与使用过程中有任何意见或建议，请随时与我们联系。

Email: [mcu@hdsc.com.cn](mailto:mcu@hdsc.com.cn)

网址: [www.hdsc.com.cn](http://www.hdsc.com.cn)

通信地址: 上海市张江高科园区碧波路 572 弄 39 号

邮编: 201203

---

