

## 32 位微控制器

# HC32L15 系列的时钟控制模块

### 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>简介 .....</b>	<b>3</b>
<b>3</b>	<b>HC32L15 系列的时钟模块.....</b>	<b>3</b>
3.1	简介 .....	3
3.2	说明 .....	3
3.3	功能 .....	4
3.3.1	时钟源的打开 .....	4
3.3.2	时钟源的切换.....	6
3.3.3	时钟分频控制.....	9
3.3.4	时钟中断.....	11
<b>4</b>	<b>样例代码 .....</b>	<b>12</b>
4.1	代码介绍 .....	12
4.2	代码运行 .....	13
<b>5</b>	<b>总结 .....</b>	<b>15</b>
<b>6</b>	<b>版本信息 &amp; 联系方式 .....</b>	<b>16</b>

# 1 摘要

本篇应用笔记主要介绍如何使用 HC32L15 芯片中的时钟模块。主要内容有：时钟的开启、时钟的切换、时钟分频控制和时钟中断。

# 2 简介

时钟产生单元为 MCU 提供不同类型的时钟源，由此 MCU 才能工作。

# 3 HC32L15 系列的时钟模块

## 3.1 简介

华大 HC32L15 系列单片机内部集成时钟模块，用于给芯片所有外设提供时钟信号。

## 3.2 说明

时钟源是 MCU 内部时钟和外部时钟的总称。芯片时钟源有以下四种类型：

- 外部高速时钟（HSXT）
- 外部低速时钟（LSXT）
- 内部高速 RC 时钟（HICR）
- 内部低速 RC 时钟（LICR）

MCU 选择以上其中一种，作为主时钟。

主时钟经过基础分频后，称为基础时钟，也是 CPU 的工作时钟（HCLK）。

基础时钟再经过总线分频，得到各个总线的时钟频率。包括 APB0 总线时钟（PCLK0）和 APB1 总线时钟（PCLK1）

## 3.3 功能

### 3.3.1 时钟源的打开

系统上电后，HICR 和 LICR 默认打开，并且，LICR 不可关闭。HSXT 和 LSXT 默认关闭。

注意：

1. 打开一个时钟，并不会替换当前时钟作为主时钟。只是指示其工作状态。
2. 在低速时钟（LICR/LSXT）为主时钟时，无法再同时打开高速时钟(HSXT/HICR)。即，切换到低速时钟后，高速时钟会被自动关闭。

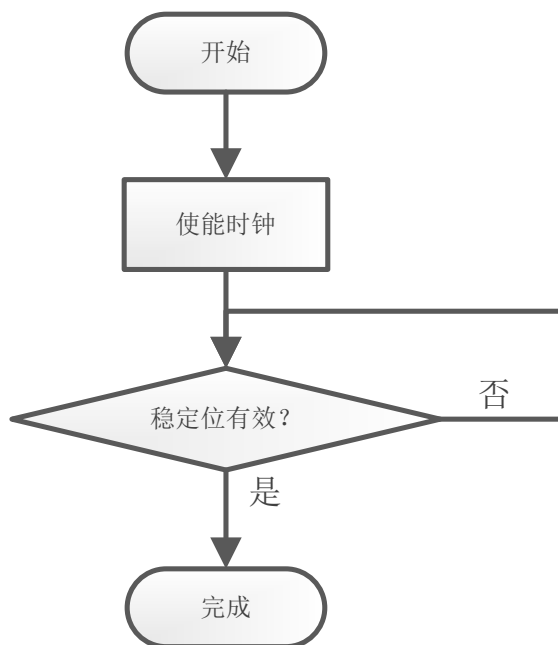


图 1 时钟源打开流程

### 3.3.1.1 时钟打开步骤

#### 使能时钟

如果是 HSXT，需要先使能 BGR 功能。（M0P\_WSUNIT->BGR\_CTL\_f.BGR\_EN = 1;）设置时钟源对应的寄存器使能位为 1。

相关寄存器：CTRL.SUB\_EN / CTRL.MAIN\_EN / CTRL.HC\_EN

如果是 HSXT，需设置相应的时钟频率（M0P\_OSC->HSXT\_PWRCTL\_f.HXSEL）

参考值如下：

HSXT	4M~6M	6M~12M	12M~20M	20M~32M
HXSEL	0	1	2	3

关寄存器：M0P\_OSC->HSXT\_PWRCTL\_f.HXSEL

#### 稳定位有效

查询时钟稳定位，为 1 时说明时钟源已打开。

相关寄存器：CLK\_STB.SUB\_RDY / CLK\_STB.MAIN\_RDY / CLK\_STB.HC\_RDY

例如在 HICR 时钟源下，开启 HSXT，过程如下：

```
M0P_WSUNIT->BGR_CTL_f.BGR_EN = 1;
M0P_OSC->HSXT_PWRCTL_f.HXSEL = 1;

M0P_CLOCK->CTRL_f.MAIN_EN = 1;

while (FALSE == M0P_CLOCK->CLK_STB_f.MAIN_RDY);
```

### 3.3.2 时钟源的切换

HC32L15 有四个时钟源输入：HICR, LICR, HSXT, LSXT。四个输入可以任意切换。但在 LICR 向 HSXT 切换时，步骤会有不同。

#### 3.3.2.1 HICR /HSXT 切换其他时钟源

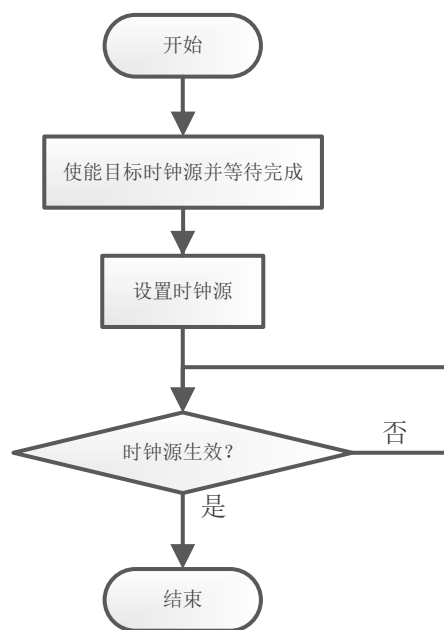


图 2 过程

切换流程分成三步：

1. 使能目标时钟源。（方法参考 时钟打开章节）
2. 设置时钟源

相关寄存器 CTRL.INPUT

3. 等待新时钟源生效

当 CLK\_STB.SRC 和 CTRL.INPUT 的值一致时，说明新时钟源生效。

相关寄存器 CLK\_STB.SRC

例：HICR 切 HSXT

```
// 使能时钟
M0P_WSUNIT->BGR_CTL_f.BGR_EN = 1;
M0P_OSC->HSXT_PWRCTL_f.HXSEL = 1;

M0P_CLOCK->CTRL_f.MAIN_EN = 1;
while (FALSE == M0P_CLOCK->CLK_STB_f.MAIN_RDY);

// 设置新时钟源
M0P_CLOCK->CTRL_f.INPUT = 0x1;

// 等待新时钟源生效
while (0x1 != M0P_CLOCK->CLK_STB_f.SRC);
```

### 3.3.2.2 LICR /LSXT 切换低速时钟

方法同 3.3.2.1

### 3.3.2.3 LICR /LSXT 切换 HICR

方法同 3.3.2.1。但可以省略步骤 1（使能 HICR 的过程）

### 3.3.2.4 LICR /LSXT 切换 HSXT

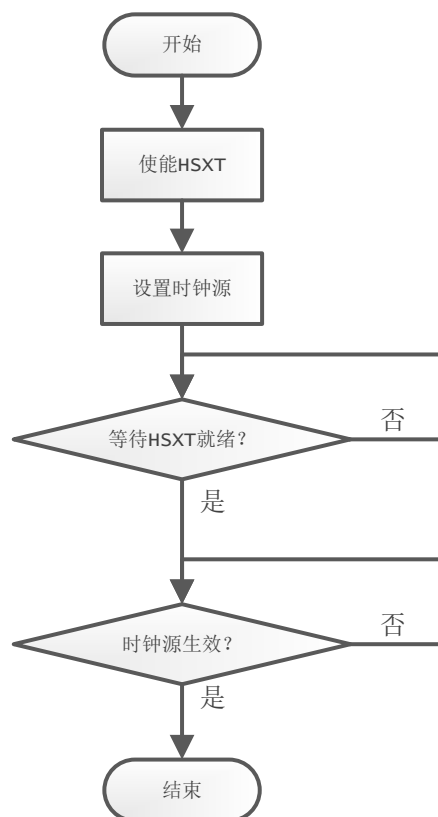


图 3 系统时钟切换流程（低速-高速）

此切换流程和 3.3.2.1 的区别是，使能 HSXT 后，先设置时钟源，再等待 HSXT 使能完成。

例：LSXT 切换 HSXT

```
// 使能时钟
M0P_WSUNIT->BGR_CTL_f.BGR_EN = 1;
M0P_OSC->HSXT_PWRCTL_f.HXSEL = 1;
M0P_CLOCK->CTRL_f.MAIN_EN = 1;

// 设置新时钟源
M0P_CLOCK->CTRL_f.INPUT = 0x1;

// 等待HSXT完成
while (FALSE == M0P_CLOCK->CLK_STB_f.MAIN_RDY);

// 等待新时钟源生效
while (0x1 != M0P_CLOCK->CLK_STB_f.SRC);
```



### 3.3.3 时钟分频控制

时钟控制从时钟源的输入到输出到外设，整个流程如下图所示：

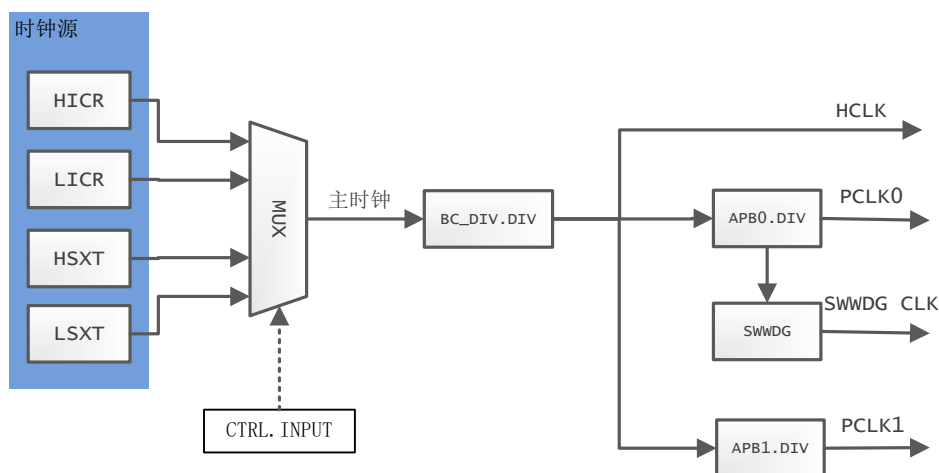


图 4 时钟源选择/分频/输出

时钟初始化设置步骤：

1. 选择时钟输入源。方法参考上一章节（时钟源的切换）。上电时，时钟源为 HICR。
2. 设置 base\_div。此分频不可屏蔽。分频范围参考手册。经过此分频后的时钟（HCLK）就是 CPU 的工作频率。
3. 外设都连在 APB 总线上，共有 2 条 APB 总线，APB0 和 APB1。外设具体分配在哪个总线上，需参考手册。

APB0 总线不可关闭。

APB1 总线默认打开，可关闭。

例 1：

Systick 使用 HCLK 作为时钟源，并可通过 base\_div 对其进行分频设置。

如已在 8M HSXT 下，base\_div 4 分频，输出 1ms 宽（500hz）的方波，如下样例为 Systick 的配置代码，省去 SysTick\_Handler 中 IO 翻转部分。

```

// base div 4 分频
M0P_CLOCK->BC_DIV_f.DIV = 3;    // 1/4
// 此时，HCLK = 8M/4 = 2M，输出1ms的Tick数为 2M/1000.
SysTick_Config(2*1000*1000/1000);
  
```

此时，程序可 1ms 一次进入 SysTick\_Handler 中断。

例 2:

如已在 8M HSXT 下，使用 CT 在 TIOA0\_0 口，输出 125KHz 的 PWM。

```
M0P_CLOCK->BC_DIV_f.DIV = 3;      // 1/4
M0P_CLOCK->APB1_CTRL_f.DIV = 1;    // 1/2
// APB1 = 8M / 4 / 2 = 1M

Gpio_SetFunc_TIOA0_0();
//CT config
M0P_CTIOSEL03->CTSEL0123_f.SEL01 = 3;
M0P_CT0_RT->RT_TCR_f.FUNC = 3;
M0P_CT0_RT->RT_TCR_f.ENCNT = 1;
M0P_CT0_RT->RT_CSR = 3;

// freq 1M/4/2 = 125K
M0P_CT0_RT->RT_TCR_f.START = 1;
while (1);
```

此时，TIOA0\_0 口输出 125Khz PWM。

### 3.3.4 时钟中断

时钟产生单元有以下中断源：

- HSXT 振荡稳定等待完成中断

HSXT 振荡稳定等待完成后，中断发生。

- LSXT 振荡稳定等待完成中断

LSXT 振荡稳定等待完成后，中断发生。

- FCS 中断（HSXT 频率异常检测）

使能 HSXT 频率异常检测功能，当检测到 HSXT 频率不在设定范围之内时发生中断。

- MCSV 中断（HSXT 失效检测）

使能 HSXT 失效检测功能，当检测到 HSXT 停止振荡时发生中断。

- SCSV 中断（LSXT 失效检测）

使能 LSXT 失效检测功能，当检测到 LSXT 停止振荡时发生中断。

例：时钟中断 HSXT 和 LSXT

这两个中断源用于提示晶振状态稳定。设定稳定时间到达后，如使能该中断，就会产生。

例 1：

在之前的时钟打开样例中，使用的是轮询的方法来查询稳定标志，也可采用中断方式来实现。环境是：HSXT = 8M, FCS 检测默认打开，则使用的是 HICR 来计数。

```
static uint8_t u8StbFlag = FALSE;
// 中断向量表的回调函数，忽略其他共享中断源
void HSXT_IrqCbK(void)
{
    u8StbFlag = TRUE;
    M0P_CLOCK->INT_CLR_f.MAIN = TRUE;
}

//=====
// 打开中断
NVIC_ClearPendingIRQ(OSC_WU_RTC_IRQn);
NVIC_EnableIRQ(OSC_WU_RTC_IRQn);
NVIC_SetPriority(OSC_WU_RTC_IRQn, DDL_IRQ_LEVEL_DEFAULT);

M0P_CLOCK->CSWT_f.MAINWT = 4;
M0P_WSUNIT->BGR_CTL_f.BGR_EN = 1;
M0P_OSC->HSXT_PWRCTL_f.HXSEL = 1;
M0P_CLOCK->CTRL_f.MAIN_EN = 1;
M0P_CLOCK->INT_CTRL_f.MAINEN = TRUE;
// 等待中断，实测8.2ms
while (FALSE == u8StbFlag);
```

## 4 样例代码

### 4.1 代码介绍

上述的工作流程可以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 DDL 的样例代码并使用其中的 clk 的 Example 进行验证。

以下部分简要介绍 clk\_setclk1 的代码(example\clk\clk\_setclk1)从 HICR 切换到 LSXT 的过程：

1) 设置时钟源：

```
if (Ok != Clk_SetSource(ClkLSXT))
{
    break;
}
```

2) 等待时钟稳定：

```
while (FALSE == Clk_GetClkRdy(ClkLSXT));
```

3) 确认时钟源已切换：

```
while (ClkLSXT != Clk_GetCurSource());
```

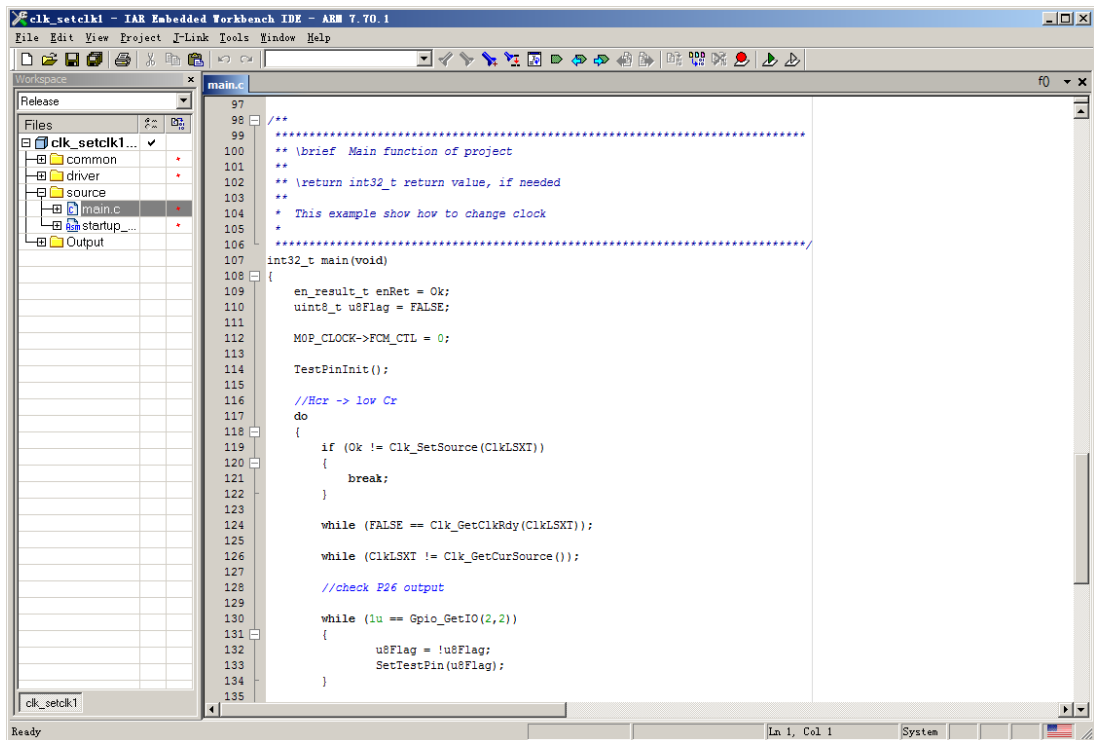
## 4.2 代码运行


用户可以通过华大半导体的网站下载到 Clk 的样例代码，并配合学习板（比如‘SK-HC32L156-64L V10’）运行相关代码学习使用时钟模块。

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

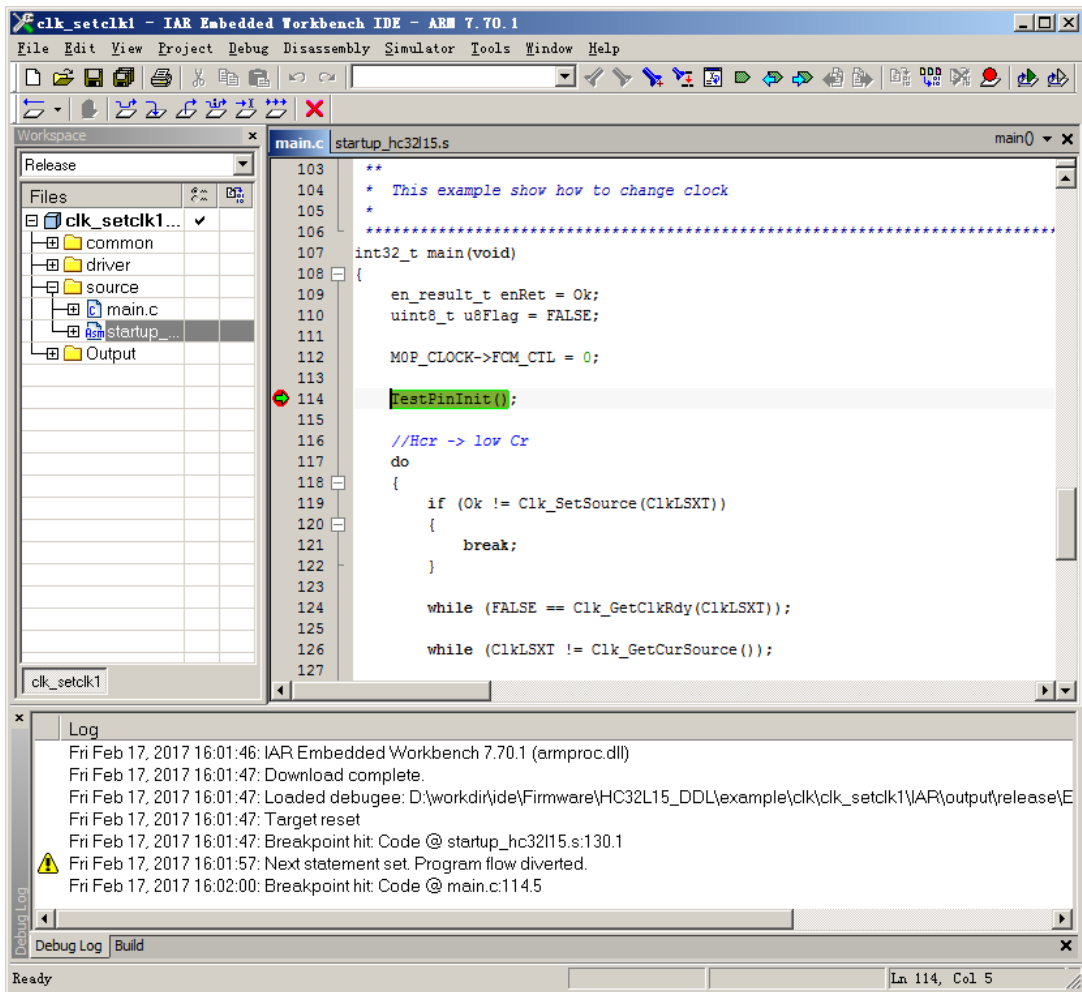
- 确认安装正确的 IAR EWARM V7.70 工具（请从 IAR 官网自行下载并安装）。
- 获取‘SK-HC32L156-64L V10’学习板。
- 从华大半导体网站下载 DDL 样例代码。
- 下载并运行样例代码（example/clk/clk\_setclk1）：

1) 打开工程项目，并打开‘main.c’如下视图：

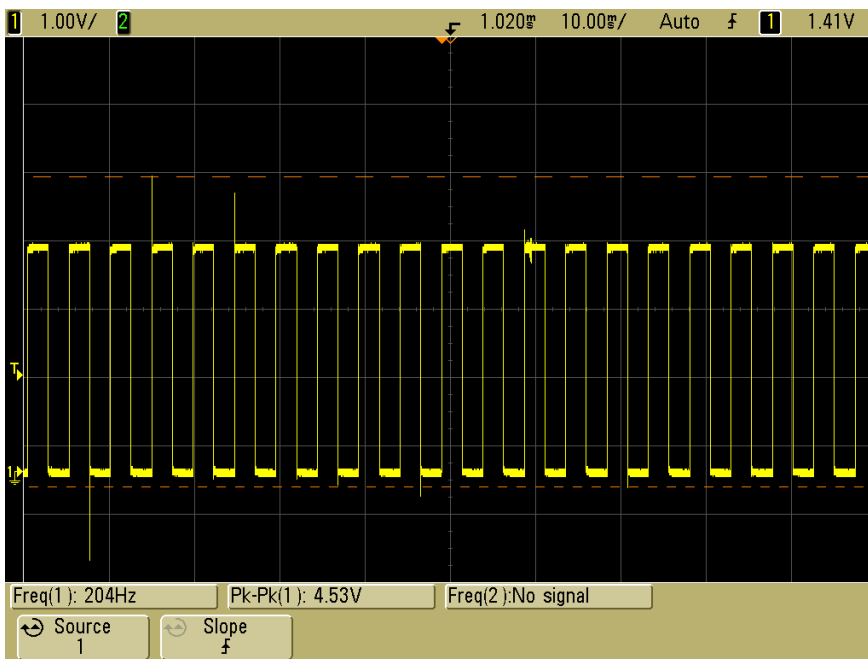


2) 点击  重新编译整个项目并将代码下载到学习板上。

3) 可以看见类似如下的视图:



4) 将学习板的 P00 连接示波器，观察波形输出:



## 5 总结

以上章节简要介绍了时钟模块，详细说明了 HC32L15 系列时钟模块并且演示了如何使用相关的样例代码进行时钟配置，在开发中用户可以根据自己的实际需要使用该模块。

## 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

---

