

利用 AT91SAM7S64 的 PCK 时钟输出引脚作为信号源

文档编号	MAN2001A_CH				
文档版本	Rev. A				
文档摘要	详细描述了利用 S64 的 PCK 时钟输出引脚作为信号源的配置过程				
关键词	AT91SAM7S64、PCK、时钟、S64 最小系统板				
创建日期	2009-11-17	创建人员	Robin	审核人员	Hotislandn
文档类型	公开发布/开发板配套文件				
版权信息	Mcuzone 原创文档，转载请注明出处				

更新历史

版本	时间	更新	作者
Rev. A	2009-11-17	初始创建	Robin

微控电子 乐微电子
杭州市登云路 639 号 2B143
销售 TEL: 86-571-89908193 13957118045
支持 TEL: 18913989166 13957118045
FAX: 86-571-89908193
www.mcuzone.com www.atarm.com

1.概述

AT91SAM7S64 具备 3 个 PCK 时钟输出引脚 (PCK0-2)，该引脚可以从 SLCK，MCK，PLLCK 这 3 个时钟源中进行选择，并且可以通过分频器分频后输出，分频器可以提供 $1/2/4/8/16/32/64$ 分频系数。这个功能可以用来给其它外设或者 MCU 提供时钟信号，或者用来作为方波信号输出。由于有 3 种时钟源可以选择，并且有多种分频系数，所以可以输出较为丰富的时钟信号。

2. PCK 时钟输出引脚的配置和实测波形

2.1 PCK 逻辑框图

先了解一下时钟源名称：

SLCK：慢时钟，S64 芯片内部的 RC 时钟，32K 左右，精度不高，但是恒定；

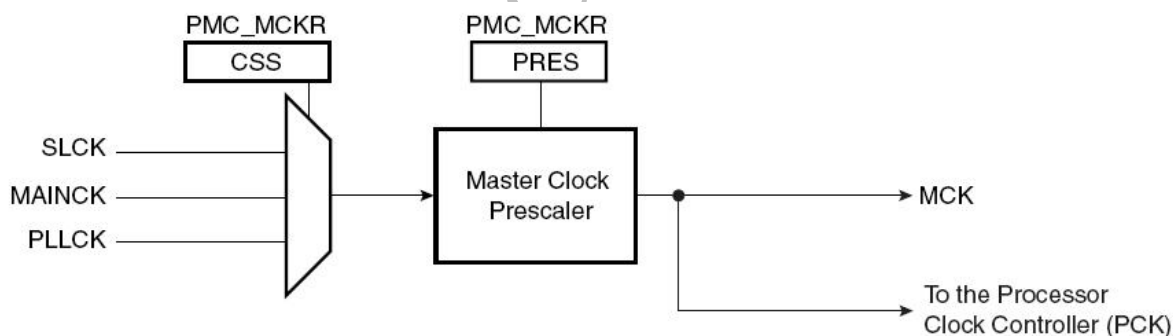
MAINCK：主时钟，即外部晶振，一般为 18.432MHz；

PLLCK：PLL 输出，从 MAINCK 经过 MUL 和 DIV 而来，范围很宽；

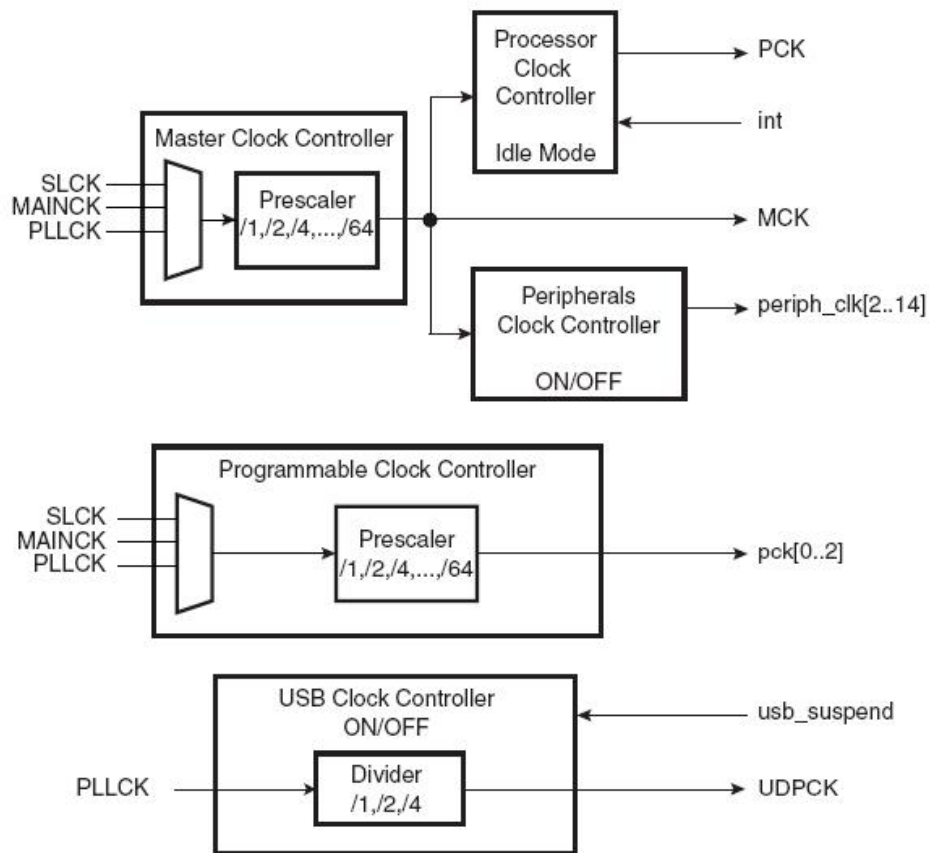
MCK：Master Clock，也叫主时钟，不过 MCK 是从 SLCK/MAINCK/PLLCK 经过 $1/2/4\dots/64$ 而来；

PCK：Processor Clock，处理器时钟，频率等同于 MCK，也是从 SLCK/MAINCK/PLLCK 经过 $1/2/4\dots/64$ 而来。

需要注意的就是要区分 MAINCK 和 MCK，还有就是区分 MCK 和 PCK。再有就是各种时钟的来源，看下面的图片就会比较清楚：



Power Management Controller Block Diagram



2.2 测试代码和实测波形

下面我们以本站的 S64 最小系统板为例进行演示。首先我们直接给出测试用代码：

```
#include <AT91SAM7S64.H> /* AT91SAM7S64 definitions */
#include "..\Board.h"

/* S64 PCK test */
/* PCK2 PA31 Peripheral B */
#define PCK2_PIN    (1 << 31)
#define PCK_CSS_SLCK    (0)
#define PCK_CSS_MCK    (1)
#define PCK_CSS_PLL    (3)
#define PCK_PRE_1    (0)
#define PCK_PRE_2    (1)
#define PCK_PRE_4    (2)
#define PCK_PRE_8    (3)
```

利用 AT91SAM7S64 的 PCK 时钟输出引脚作为信号源

MAN2001A_CH

```
#define PCK_PRE_16      (4)
#define PCK_PRE_32      (5)
#define PCK_PRE_64      (6)
/* PCK_Output(PCK_CSS_MCK, PCK_PRE_16); 48MHz / 16 = 3MHz*/
void PCK_Output(unsigned int clk_css, unsigned int clk_pres)
{
    /* set PIO as PA31 as PCK2 */
    AT91C_BASE_PIOA->PIO_PDR = PCK2_PIN;
    AT91C_BASE_PIOA->PIO_BSR = PCK2_PIN;

    /* disable PCK2 before changing setting */
    AT91C_BASE_PMC->PMC_SCDR = (1 << 10);

    /* set PCK2 */
    AT91C_BASE_PMC->PMC_PCKR[2] = (clk_pres << 2) | clk_css;

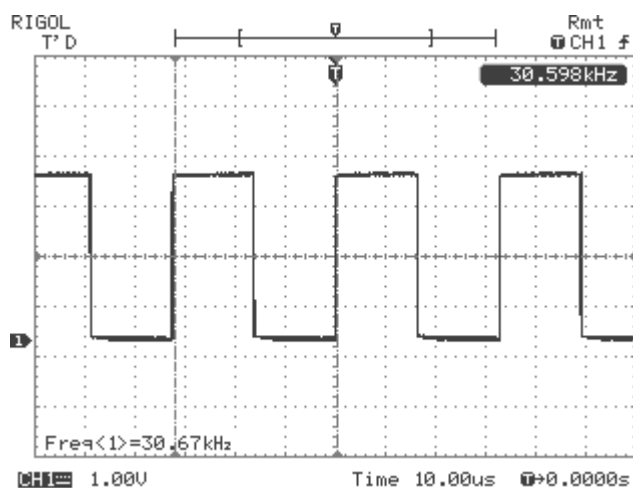
    /* enable PCK2 */
    AT91C_BASE_PMC->PMC_SCER = (1 << 10);
}

void main()
{
    PCK_Output(PCK_CSS_PLL, PCK_PRE_4);
    while(1)
    ;
}
```

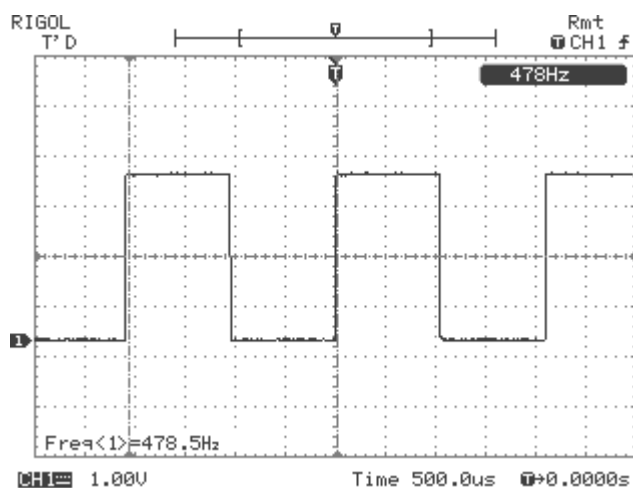
由于 Keil 和 IAR 的编译版本不断更新，为了尽可能的减少由于编译器版本变化而带来的编译问题，我们并不提供现成的工程文件，仅提供关键代码。利用该关键代码客户可以很方便的构建符合自己编译器版本的工程文件。具体做法如下：

- 1). 安装并打开 Keil 软件;
- 2). 打开 Keil 安装目录下的 AT91SAM7S 的 Blinky 例子，如：C:\Keil\ARM\Boards\Atmel\AT91SAM7S-EK;
- 3). 修改 CPU 为 AT91SAM7S64(以本站 S64 最小系统板为例)，并按照实际情况修改 debug 选项卡;
- 4). 打开 Blinky.c，删除原先代码，加入上文代码;
- 5). 重新编译并下载代码;
- 6). 观察 PA31 引脚输出的波形，如果观察到波形可以修改参数改变输出波形;

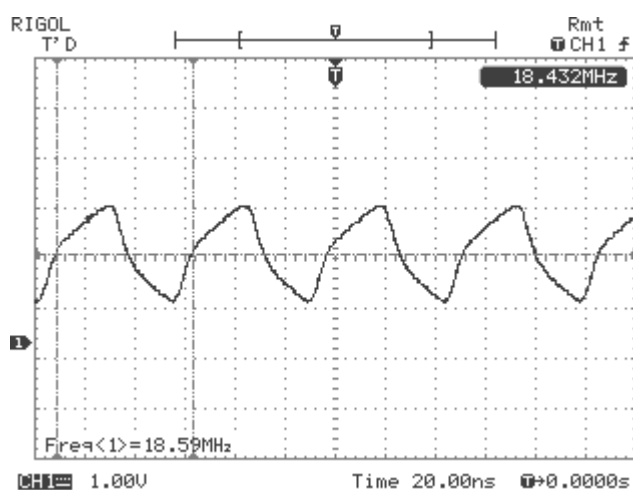
注意：以上代码设定从 PA31 输出波形；可以修改时钟源和分频系数来改变输出波形频率。



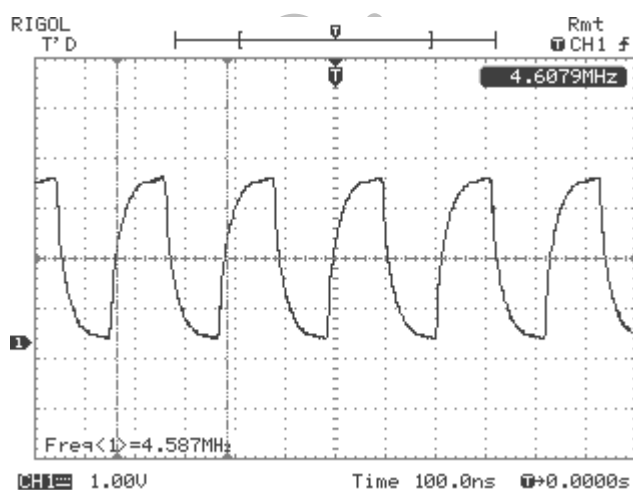
SLCK 时钟源，分频系数为 1



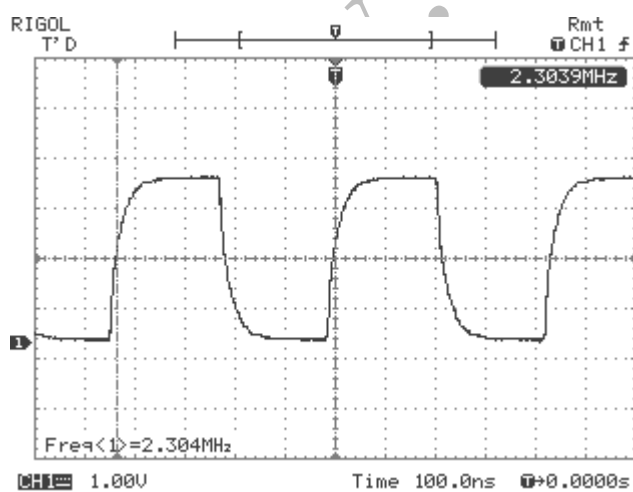
SLCK 时钟源，分频系数为 64



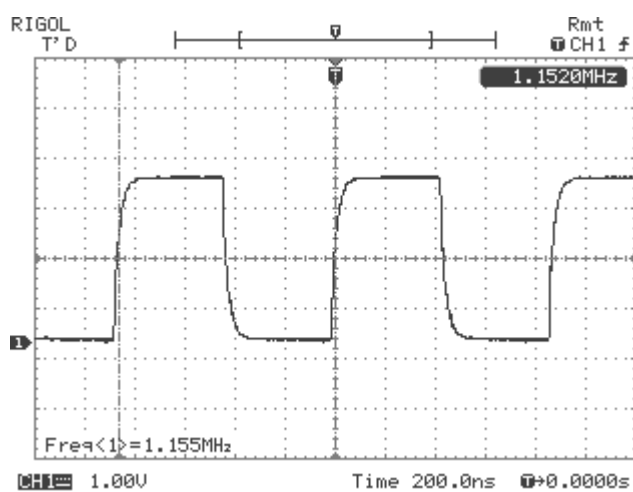
MAINCK 时钟源，分频系数为 1



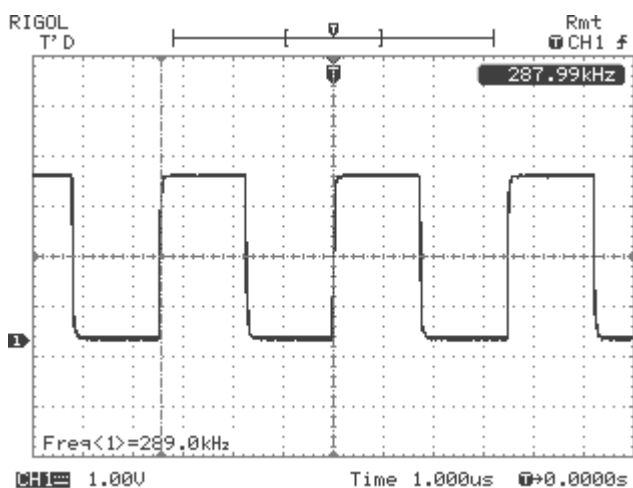
MAINCK 时钟源，分频系数为 4



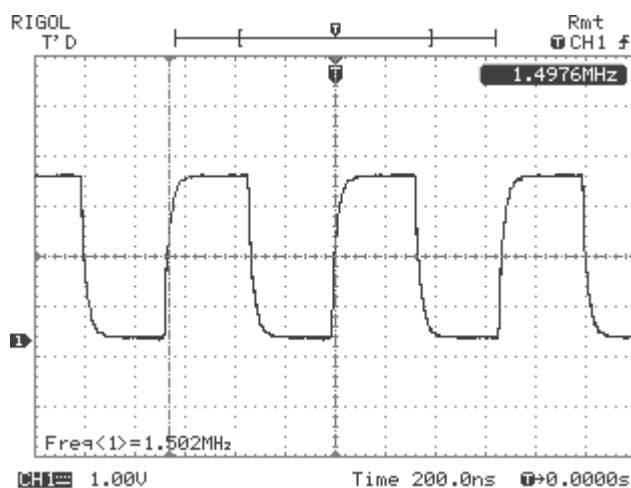
MAINCK 时钟源，分频系数为 8



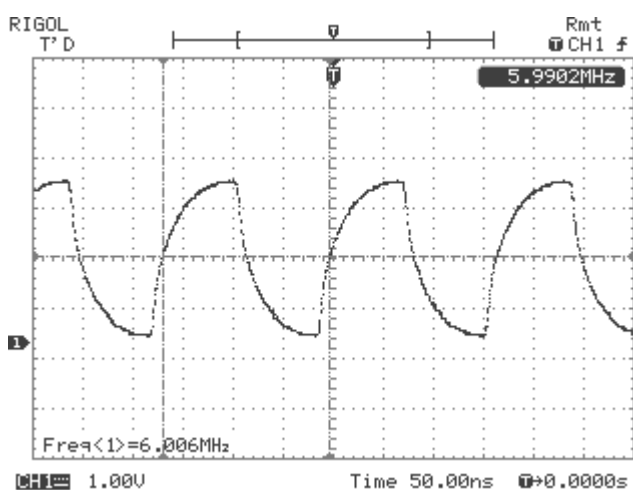
MAINCK 时钟源，分频系数为 16



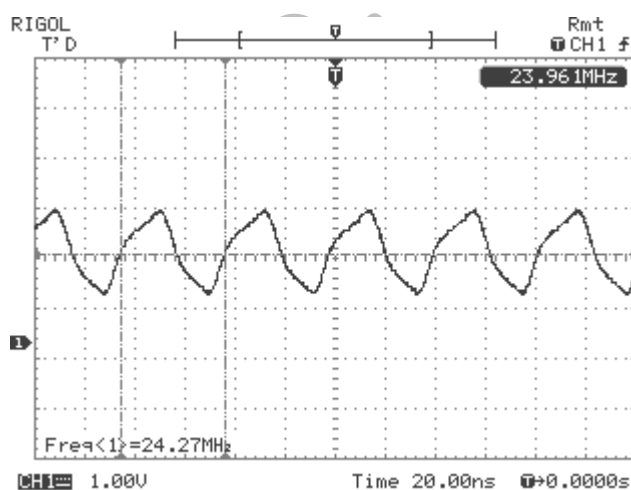
MAINCK 时钟源，分频系数为 64



PLLCK 时钟源，分频系数为 64

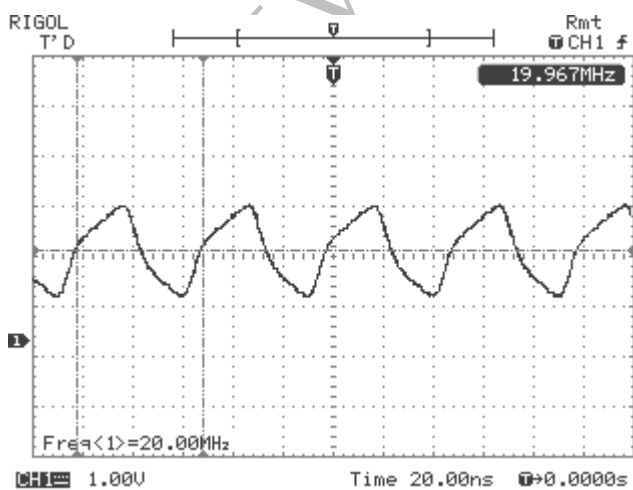


PLLCK 时钟源，分频系数为 16

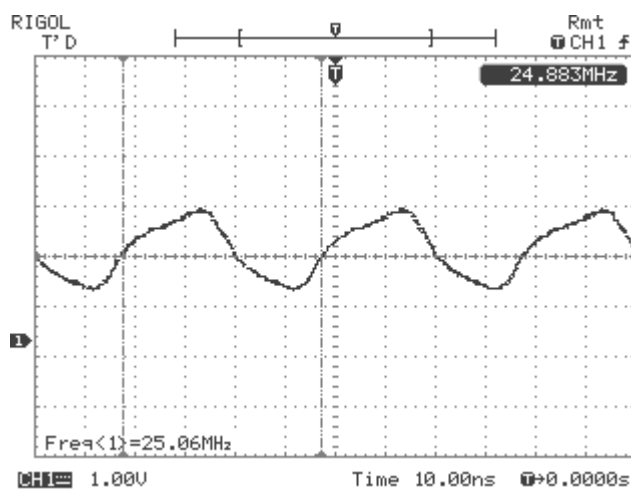


PLLCK 时钟源，分频系数为 4

注意：可以修改 DIV 和 MUL 系数来改变 PLLCK 输出频率。对于 Keil 提供的这个例子，PLL 的 MUL 系数是 5，DIV 系数是 25，即 PLLCK 输出是： $18.432\text{MHz} \times (25+1)/5 = 95.8\text{MHz}$ 。下面是修改 MUL 和 DIV 系数后的输出。



PLLCK 时钟源，分频系数为 4，DIV=6，MUL=25

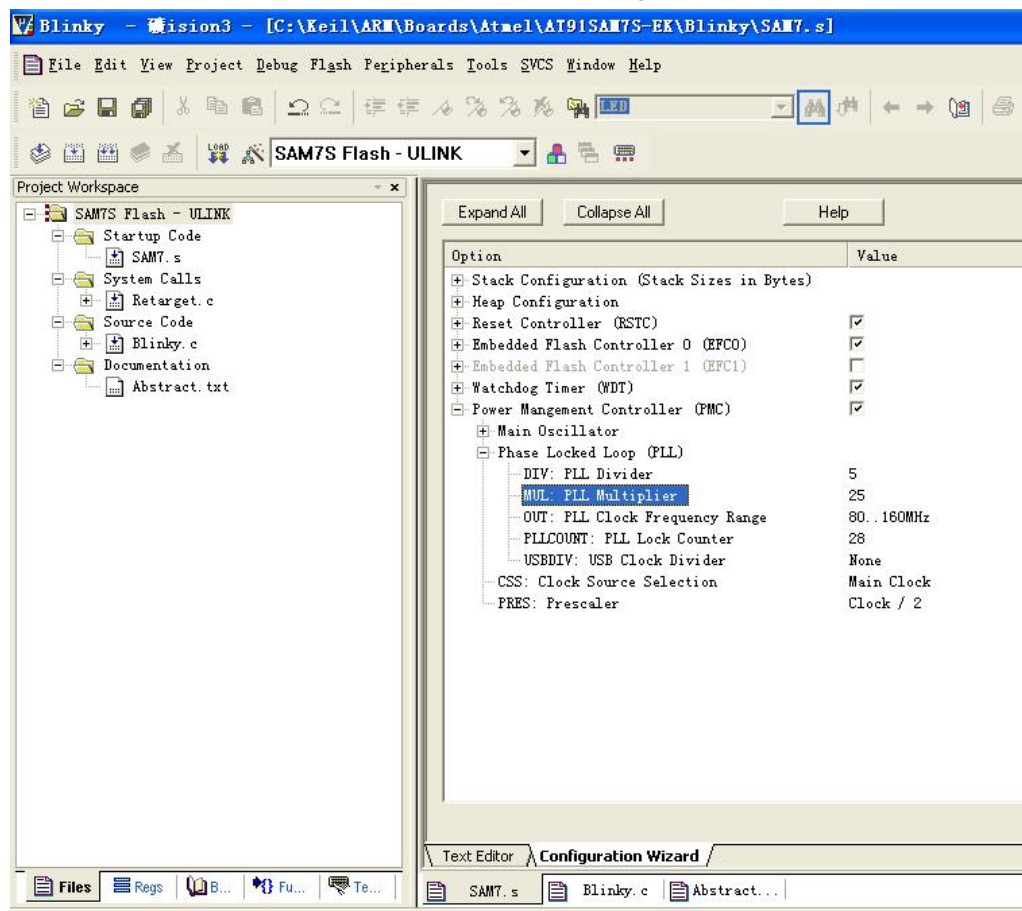


PLLCK 时钟源，分频系数为 4，DIV=5，MUL=26

利用 AT91SAM7S64 的 PCK 时钟输出引脚作为信号源

MAN2001A_CH

MUL 和 DIV 的参数设定可以方便的使用 Keil 的 configuration wizard 来操作:



配套的 S64 最小系统板

利用 AT91SAM7S64 的 PCK 时钟输出引脚作为信号源

MAN2001A_CH