

# AT91SAM7S-EK softpack 1.5 代码解读之 PWM

——基于 7S64-DEK Rev2.0 和 IAR EWARM5.40

文档编号	MAN2030A_CH				
文档版本	Rev. A				
文档摘要	基于 AT91SAM7S-EK 开发板的代码解读，PWM 代码解读				
关键词	AT91SAM7S-EK, IAR EWARM5.40, PWM, J-Link				
创建日期	2010-08-04	创建人员	Cust126	审核人员	Robin
文档类型	公开发布/开发板配套文件				
版权信息	<a href="#">Mcuzone</a> 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	2010-08-04	初始创建	Cust126

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](http://www.mcuzone.com) [www.atarm.com](http://www.atarm.com)

# 1. 概述

本文档以 7S64-DEK 为硬件平台，IAR EWARM 为编译器平台，使用 J-Link 作为调试工具，演示并解读 AT91SAM7S64 的 PWM 操作流程。

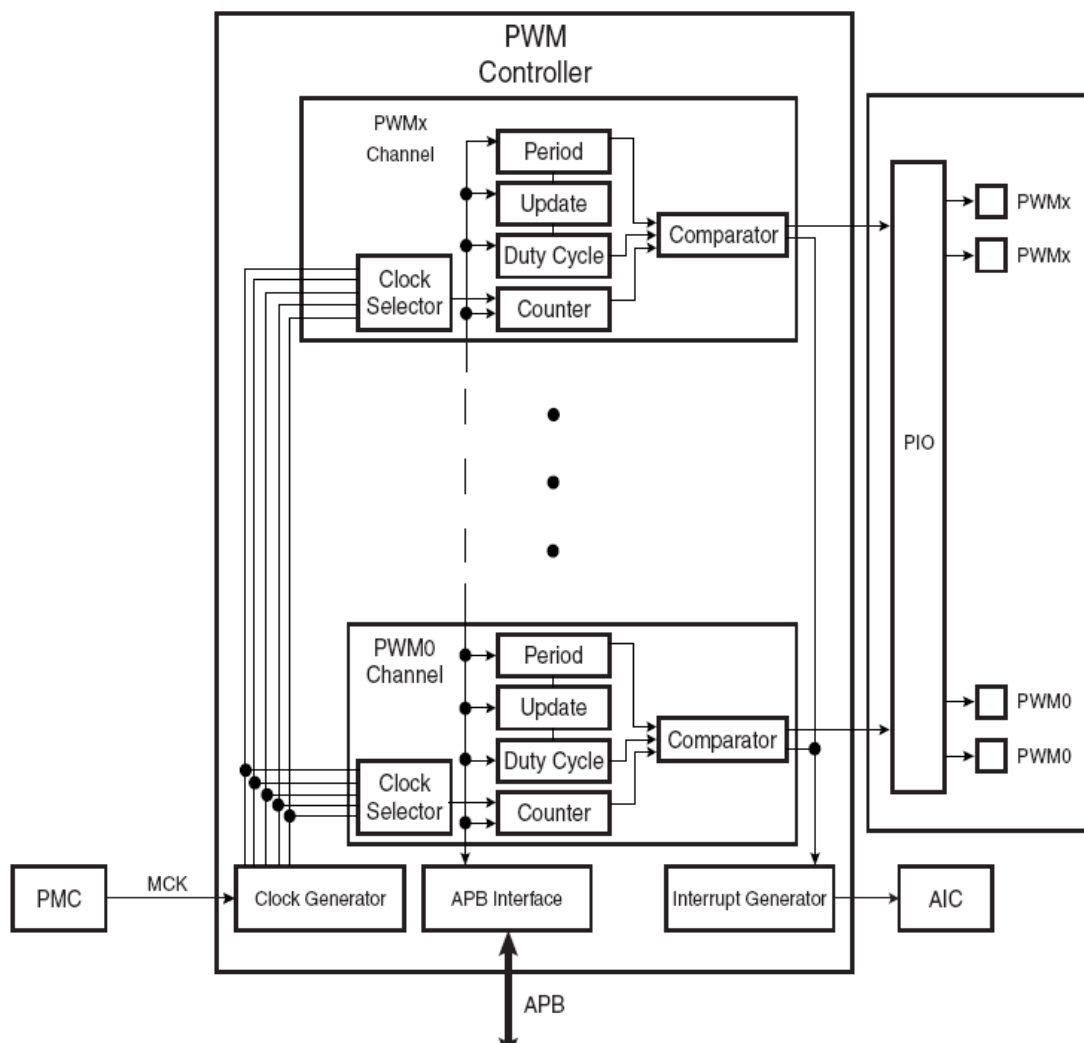


## 2. PWM 操作

### 2.1 PWM 操作流程介绍

PWM 宏单元独立控制几个通道。每个通道控制一个输出方波，通过用户可以配置输出波形的周期、占空比、极性。每个通道选择并使用一个有时钟产生器提供的时钟。时钟发生器提供几个由 PWM 宏单元分频时钟后得到的时钟。

PWM 控制方框图：



## 2.2 PWM 操作的目的与功能描述

AT91 softpack 1.5 的 SAM7S-EK 包内的 basic-pwm-project 工程目的主要是帮助初学者熟悉 PWM 在 AT91SAM7S 系列上的用法。功能是使两个 LED 以不同的频率逐渐的亮和灭。

## 2.3 示例代码

以下代码截取自 AT91 softpack 1.5 的 SAM7S-EK 包内的 basic-pwm-project ， 基于 IAR EWARM 平台

下面对 basic-adc-project 的主要代码进行注释解读，首先是 main 函数内容：

```
int main(void)
{
    PIO_Configure(pins, PIO_LISTSIZE(pins));
    TRACE_CONFIGURE(DBGU_STANDARD, 115200, BOARD_MCK);
```

## McuzoneApplication Notes

```
printf("-- Basic PWMC Project %s --\n\r", SOFTPACK_VERSION);
printf("-- %s\n\r", BOARD_NAME);
printf("-- Compiled: %s %s --\n\r", __DATE__, __TIME__);

UTIL_WaitTimeInMs(BOARD_MCK, 1000);
UTIL_WaitTimeInUs(BOARD_MCK, 1000);

// Enable PWMC peripheral clock          //使能 PWM 时钟
AT91C_BASE_PMC->PMC_PCER = 1 << AT91C_ID_PWMC;

// Settings:
// - 100kHz PWM period (PWM_FREQUENCY)
// - 1s rise/fall time for the LED intensity

// Set clock A to run at 100kHz * MAX_DUTY_CYCLE (clock B is not used)
PWMC_ConfigureClocks(PWM_FREQUENCY * MAX_DUTY_CYCLE, 0, BOARD_MCK);

// Configure PWMC channel for LED0 (left-aligned)      //配置 PWM 通道 0 左对齐
PWMC_ConfigureChannel(CHANNEL_PWM_LED0, AT91C_PWMC_CPRE_MCKA, 0, 0);
PWMC_SetPeriod(CHANNEL_PWM_LED0, MAX_DUTY_CYCLE);
PWMC_SetDutyCycle(CHANNEL_PWM_LED0, MIN_DUTY_CYCLE);

// Configure PWMC channel for LED1 (center-aligned, inverted polarity) //配置 PWM 通道 1
PWMC_ConfigureChannel(CHANNEL_PWM_LED1, AT91C_PWMC_CPRE_MCKA, AT91C_PWMC_CALG,
AT91C_PWMC_CPOL);
PWMC_SetPeriod(CHANNEL_PWM_LED1, MAX_DUTY_CYCLE);
PWMC_SetDutyCycle(CHANNEL_PWM_LED1, MIN_DUTY_CYCLE);

// Configure interrupt on channel #1
AIC_ConfigureIT(AT91C_ID_PWMC, 0, ISR_Pwmc);
AIC_EnableIT(AT91C_ID_PWMC);
PWMC_EnableChannelIt(CHANNEL_PWM_LED0);

// Enable channel #1 and #2
PWMC_EnableChannel(CHANNEL_PWM_LED0);
PWMC_EnableChannel(CHANNEL_PWM_LED1);

// Infinite loop
while (1);
}
```

## McuzoneApplication Notes

### 2.3.2 重要子函数代码注释解读

以下是对几个重要的子函数进行分析：

```
//-----  
/// Interrupt handler for the PWM controller. Counts the number of periods and  
/// updates the duty cycle after a given number.  
//-----  
static void ISR_Pwmc(void)  
{  
    static unsigned int count = 0;  
    static unsigned int duty = MIN_DUTY_CYCLE;  
    static unsigned char fadeIn = 1;  
  
    // Interrupt on channel #1  
    if ((AT91C_BASE_PWMC->PWMC_ISR & AT91C_PWMC_CHID1) == AT91C_PWMC_CHID1) {  
  
        count++;  
  
        // Fade in/out  
        if (count == (PWM_FREQUENCY / (MAX_DUTY_CYCLE - MIN_DUTY_CYCLE))) {  
  
            // Fade in  
            if (fadeIn) {  
  
                duty++;  
                if (duty == MAX_DUTY_CYCLE) {  
  
                    fadeIn = 0;  
                }  
            }  
            // Fade out  
            else {  
  
                duty--;  
                if (duty == MIN_DUTY_CYCLE) {  
  
                    fadeIn = 1;  
                }  
            }  
        }  
  
        // Set new duty cycle  
        count = 0;  
        PWMC_SetDutyCycle(CHANNEL_PWM_LED0, duty);  
    }  
}
```

## McuzoneApplication Notes

```
        PWMC_SetDutyCycle(CHANNEL_PWM_LED1, duty);  
    }  
}  
}
```

### 2.2.3 运行结果

LED2 和 LED3 以一定的频率交替的渐亮和渐灭。

www.mcuzone.com