

# UM324xF 时钟配置指南

版本：V1.1



广芯微电子（广州）股份有限公司

<http://www.unicmicro.com/>

## 目录

1	概述.....	1
2	时钟软件配置.....	2
2.1	芯片时钟源配置.....	2
2.2	系统时钟源配置.....	2
2.3	系统时钟配置.....	3
2.4	外设总线时钟配置.....	4
2.5	其它配置.....	6
3	相关变量.....	7
4	版本修订.....	8

# 1 概述

系统存在 4 个时钟源：

- 一个频率为48MHz的高精度内部时钟RCH
- 一个频率为32KHz的内部时钟RCL
- 一个频率为32.768KHz的外部晶体时钟XTL
- 支持外部晶体时钟XTH

内置 2 个 PLL，提供系统时钟、音频时钟和 USB 时钟。

时钟控制单元提供了一系列频率的时钟功能，包括一个内部 48M RC 振荡器时钟(RCH)、一个外部高速晶体振荡器时钟(XTH)、一个内部 32K RC 振荡器时钟(RCL)、一个外部低速晶体振荡器时钟(XTL)、两个锁相环(PLL)、一个 XTH 时钟监视器、时钟预分频器、时钟多路复用器和时钟门控电路。

AHB、APB、Cortex™-M4 源自系统时钟(SYS\_CLK)，系统时钟的时钟源可以选择 XTH、RCH、PLL 或低频 32KHz 的 RCL/XTL 时钟。独立看门狗定时器有采用低频的时钟源（RCL 或者 XTL），实时时钟(RTC)使用 RCL 或 XTL 作为时钟源。

注意：

- 如需要支持音频，可配置PLL小数分频。
- 对EMC要求严格应用（电机等），可打开PLL扩频功能。

## 2 时钟软件配置

### 2.1 芯片时钟源配置

在“um324xF\_hal\_conf.h”头文件中，设置RCH宏定义和XTH宏定义。

- RCH\_VALUE。
- XTH\_VALUE。

这两个宏定义了内部高速时钟频率和外部晶振频率大小。其中RCH固定为48MHz，XTH根据实际需求更改，UM324xF\_EVB核心板上外部晶振为12MHz。

注：推荐外部晶振选择12MHz、24MHz和48MHz，便于倍频和分频配置，用于给USB提供准确的48MHz时钟。

代码如下所示：

```
#if !defined (RCH_VALUE)
    #define RCH_VALUE    (48000000U) /*!< Value of the Internal oscillator in Hz*/
#endif /* RCH_VALUE */

#if !defined (XTH_VALUE)
    #define XTH_VALUE    (12000000U) /*!< Value of the External oscillator in Hz */
#endif /* XTH_VALUE */
```

### 2.2 系统时钟源配置

在“um324xF\_hal\_conf.h”头文件中，设置系统时钟源宏定义SYSCLK\_SRC，可选择以下四种：

- SYSCLK\_USE\_RCH，RCH做系统时钟源。
- SYSCLK\_USE\_XTH，XTH做系统时钟源。
- SYSCLK\_USE\_RCH\_PLL，RCH做PLL0时钟源，PLL0输出做系统时钟源。
- SYSCLK\_USE\_XTH\_PLL，XTH做PLL0时钟源，PLL0输出做系统时钟源。

SDK用例代码中，RCH做时钟源时，系统时钟固定为48MHz。XTH做时钟源时，系统时钟大小取决于外部晶振频率大小。例如：UM324xF\_EVB核心板上外部晶振为12MHz，则系统时钟为12MHz。

RCH或XTH做PLL0时钟源，PLL0输出做系统时钟源，可根据实际需求选择多种系统时钟频率。用例中默认选择XTH做PLL0时钟源，PLL0输出做系统时钟源，即SYSCLK\_SRC定义为SYSCLK\_USE\_XTH\_PLL。

代码如下所示：

```

#define SYSCLK_USE_RCH      0           //RCH as system clock(48MHz)
#define SYSCLK_USE_XTH      1           //XTH as system clock(The frequency
depends on the external crystal oscillator)
#define SYSCLK_USE_RCH_PLL  2           //RCH as the source clock of PLL0,
the system clock frequency can be given in the macro definition FCLK
#define SYSCLK_USE_XTH_PLL  3           //XTH as the source clock of PLL0,
the system clock frequency can be given in the macro definition FCLK

#define SYSCLK_SRC          SYSCLK_USE_XTH_PLL //For USB
applications, SYSCLK_USE_XTH_PLL is recommended (board level requires external
12Mhz crystal)

```

## 2.3 系统时钟配置

在“um324xF\_hal\_rcm.h”头文件中，有如下代码块结构体定义。该结构体定义了振荡器类型、XTH的状态、XTL的状态、RCH的状态、RCL的状态和PLL结构体。

```

typedef struct
{
    uint32_t OscillatorType;           /*!< The oscillators to be configured. This parameter can
be a value of @ref RCM_Oscillator_Type */
    uint32_t XTHState;                 /*!< The new state of the XTH. This parameter can be a
value of @ref RCM_XTH_Config */
    uint32_t XTLState;                 /*!< The new state of the XTL. This parameter can be a
value of @ref RCM_XTL_Config */
    uint32_t RCHState;                 /*!< The new state of the RCH. This parameter can be a
value of @ref RCM_RCH_Config */
    uint32_t RCLState;                 /*!< The new state of the RCL. This parameter can be a
value of @ref RCM_RCL_Config */
    RCM_PLLInitTypeDef PLL;           /*!< PLL structure parameters */
}RCM_OscInitTypeDef;

```

其中PLL结构体定义了PLL的状态、PLL时钟源、PLLM系数、PLLN系数和PLLP系数。

```

typedef struct
{
    uint32_t PLLState;                 /*!< The new state of the PLL. This parameter can be a value of
@ref RCM_PLL_Config */
    uint32_t PLLSource;                /*!< RCM_PLLSource: PLL entry clock source. This parameter must
be a value of @ref RCM_PLL_Clock_Source */

```

```

uint32_t PLLM;          /*!< PLLM: Division factor for PLL VCO input clock. This parameter
must be a number between Min_Data = 0 and Max_Data = 63 */
uint32_t PLLN;          /*!< PLLN: Multiplication factor for PLL VCO output clock. This
parameter must be a number between Min_Data = 16 and Max_Data = 1024 */
uint32_t PLLP;          /*!< PLLP */
}RCM_PLLInitTypeDef;

```

以系统时钟168MHz为例，参考时钟源选择12MHz的外部时钟XTH，建议通过PLL0倍频到336MHz， $F_{clk}=(F_{ref} \cdot DN)/(DP \cdot DM) = 336\text{MHz}$ ，Fclk为PLL0输出，Fref为参考时钟。该系统时钟下ATIMER和GTIMER时钟源可选择系统时钟168MHz或PLL0时钟336MHz，即ATIMER和GTIMER想使用PLL0作时钟源时，必须满足系统时钟=PLL0/2的关系，且PLL0时钟最高不超过336MHz。

代码如下所示，振荡器类型设置为外部时钟XTH作为参考时钟，开启XTH和PLL功能，设置XTH作为PLL的源时钟，设置M、N、P系数。需要注意的是，N值必须大于等于16。

```

/* Enable XTH Oscillator and activate PLL with HSE as source */
RCM_OscInitStruct.OscillatorType = RCM_OSCILLATORTYPE_XTH;
RCM_OscInitStruct.XTHState = RCM_XTH_ON;
RCM_OscInitStruct.PLL.PLLState = RCM_PLL_ON;
RCM_OscInitStruct.PLL.PLLSource = RCM_PLLSOURCE_XTH;
RCM_OscInitStruct.PLL.PLLM = 1;
RCM_OscInitStruct.PLL.PLLN = 28;
RCM_OscInitStruct.PLL.PLLP = 1; //XTH=12M ,PLL CLKOUT(Fclk)=336Mhz, Fref= XTH
= 12M , Fclk=(Fref*DN)/(DP*DM)

/*Initializes the RCM Oscillators*/
HAL_RCM_OscConfig(&RCM_OscInitStruct);

```

注：以上系统时钟对应的情况为目前SDK用例代码中所配置的情况，用户可根据实际需求更改系统时钟以及系统时钟和PLL0的分频关系。

## 2.4 外设总线时钟配置

在“um324xF\_hal\_rcm.h”头文件中，有如下代码块结构体定义。该结构体定义了时钟类型、系统时钟源、RCH分频、AHB分频、APB0分频、APB1分频、APB2分频、APB3分频、USB和SDIO分频。

```

typedef struct
{
    uint32_t ClockType;          /*!< The clock to be configured. This parameter can be a
value of @ref RCM_System_Clock_Type */

```

```

uint32_t SYSClkSource;          /*!< The clock source (SYSClKS) used as system
clock. This parameter can be a value of @ref RCM_System_Clock_Source */
uint32_t RCHDivider;           /*!< The clock RCH divider. This parameter can be a
value of @ref RCM_RCH_Clock_Source */
uint32_t AHBCLKDivider;        /*!< The AHB clock (HCLK) divider. This clock is
derived from the system clock (SYSClK). This parameter can be a value of @ref
RCM_AHB_Clock_Source */
uint32_t APB0CLKDivider;        /*!< The APB0 clock (PCLK0) divider. This clock is
derived from the AHB clock (HCLK). This parameter can be a value of @ref
RCM_APB0_Clock_Source */
uint32_t APB1CLKDivider;        /*!< The APB1 clock (PCLK1) divider. This clock is
derived from the AHB clock (HCLK). This parameter can be a value of @ref
RCM_APB1_Clock_Source */
uint32_t APB2CLKDivider;        /*!< The APB2 clock (PCLK2) divider. This clock is
derived from the AHB clock (HCLK). This parameter can be a value of @ref
RCM_APB2_Clock_Source */
uint32_t APB3CLKDivider;        /*!< The APB3 clock (PCLK3) divider. This clock is
derived from the AHB clock (HCLK). This parameter can be a value of @ref
RCM_APB3_Clock_Source */
uint32_t USBSdioClkDivider;     /*!< The USB/SDIO clock (48M) divider. This clock
is derived from the system clock (SYSClK). This parameter can be a value of @ref
RCM_USB_SDIO_Clock_Source
*/
}RCM_ClkInitTypeDef;

```

需要注意的是，APB0时钟频率不能大于102MHz，USB时钟频率必须为48MHz，SDIO时钟频率不能超过52MHz。

以上分频值可根据实际需求做相应更改。以336MHz倍频时钟、168MHz系统时钟为例，代码如下所示：

```

/* Select PLL as system clock source and configure the HCLK, PCLK0, PCLK1 ,PCLK2, and
PCLK3 clocks dividers */
RCM_ClkInitStruct.ClockType = (RCM_CLOCKTYPE_SYSClK | RCM_CLOCKTYPE_HClK
| RCM_CLOCKTYPE_PClK0 |RCM_CLOCKTYPE_PClK1 | RCM_CLOCKTYPE_PClK2|
RCM_CLOCKTYPE_PClK3);
RCM_ClkInitStruct.SYSClKSource = RCM_SYSClKSOURCE_PLL0ClK;
RCM_ClkInitStruct.AHBClKDivider = RCM_SYSClK_DIV2;    //SystemCoreClock =
Fclk/RCM_SYSClK_DIV2 = 336Mhz/2 = 168MHz
RCM_ClkInitStruct.APB0ClKDivider = RCM_PClK0_DIV4;    //APB0Clock =
SystemCoreClock/RCM_PClK0_DIV4 = 168MHz/4 = 42MHz

```

```

RCM_ClkInitStruct.APB1CLKDivider = RCM_PCLK1_DIV1;    //APB1Clock =
SystemCoreClock/RCM_PCLK1_DIV1 = 168MHz/1 = 168MHz
RCM_ClkInitStruct.APB2CLKDivider = RCM_PCLK2_DIV1;    //APB2Clock =
SystemCoreClock/RCM_PCLK2_DIV1 = 168MHz/1 = 168MHz
RCM_ClkInitStruct.APB3CLKDivider = RCM_PCLK3_DIV4;    //APB3Clock =
SystemCoreClock/RCM_PCLK3_DIV4 = 168MHz/4 = 42MHz

/*Initializes the CPU, AHB and APB busses clocks */
HAL_RCM_ClockConfig(&RCM_ClkInitStruct, FLASH_RWAITCYC_3);

```

## 2.5 其它配置

UM324xF可以选择LDO01的电压值为1.1V或0.9V，这个电压会影响不同系统频率下的EFC的读等待时间设置，读等待时间宏定义如下：

```

#define FLASH_RWAITCYC_0          FLASH_EFC_TIME_RWAITCYC_0WC
/*!< FLASH Zero Read wait cycle */
#define FLASH_RWAITCYC_1          FLASH_EFC_TIME_RWAITCYC_1WC
/*!< FLASH One Read wait cycle */
#define FLASH_RWAITCYC_2          FLASH_EFC_TIME_RWAITCYC_2WC
/*!< FLASH Two Read wait cycles */
#define FLASH_RWAITCYC_3          FLASH_EFC_TIME_RWAITCYC_3WC
/*!< FLASH Three Read wait cycles */
#define FLASH_RWAITCYC_4          FLASH_EFC_TIME_RWAITCYC_4WC
/*!< FLASH Four Read wait cycles */
#define FLASH_RWAITCYC_5          FLASH_EFC_TIME_RWAITCYC_5WC
/*!< FLASH Five Read wait cycles */

```

具体的电压档位和EFC读等待时间关系可参考《UM324xF 用户手册》中的存储系统章节。

EFC读等待时间设置可通过以下接口的第二个入口参数传入设置：

```

HAL_RCM_ClockConfig(&RCM_ClkInitStruct, FLASH_RWAITCYC_3);

```



### 3 相关变量

在“um324xF\_hal\_rcm.h”文件中，定义有如下的函数。从上往下功能分别为获取HCLK、APB0、APB1、APB2、APB3的时钟频率。在更改了时钟配置后，可通过这些函数来重新获取总线时钟，以重新配置相关的IP设置，便于代码中应用计算。

```
uint32_t HAL_RCM_GetHCLKFreq(void);  
uint32_t HAL_RCM_GetPCLK0Freq(void);  
uint32_t HAL_RCM_GetPCLK1Freq(void);  
uint32_t HAL_RCM_GetPCLK2Freq(void);  
uint32_t HAL_RCM_GetPCLK3Freq(void);
```

## 4 版本修订

版本	日期	描述
V1.0	2022.12.05	初始版
V1.1	2023.09.05	更新“外设总线时钟配置”章节描述； 更新所有代码块内容。