# 九：SystemInit()实现的功能

花了一天的时间，总算是了解了SystemInit()函数实现了哪些功能，初学STM32，，现记录如下（有理解错误的地方还请大侠指出）：

使用的是3.5的库，用的是STM32F107VC，开发环境RVMDK4.23

我已经定义了STM32F10X\_CL，SYSCLK\_FREQ\_72MHz

**函数调用顺序：**

**startup\_stm32f10x\_cl.s（启动文件） → SystemInit() →  SetSysClock () → SetSysClockTo72()**

初始化时钟用到的RCC寄存器**复位值：**

RCC\_CR = 0x0000 xx83; RCC\_CFGR = 0x0000 0000；RCC\_CIR = 0x0000 0000; RCC\_CFGR2 = 0x0000 0000;

**SystemInit()**

在调用 SetSysClock()之前RCC寄存器的值如下（都是一些与运算，或运算，在此就不赘述了）：

RCC->CR = 0x0000 0083;  RCC->CIR = 0x00FF0000; RCC->CFGR2 = 0x00000000;至于这些寄存器都代表着什么意思，详见芯片资料RCC寄存器，该文重点不在此处；

**SetSysClock()函数如下：**

static void SetSysClock(void)

{

#ifdef SYSCLK\_FREQ\_HSE

  SetSysClockToHSE();

#elif defined SYSCLK\_FREQ\_24MHz

  SetSysClockTo24();

#elif defined SYSCLK\_FREQ\_36MHz

  SetSysClockTo36();

#elif defined SYSCLK\_FREQ\_48MHz

  SetSysClockTo48();

#elif defined SYSCLK\_FREQ\_56MHz

  SetSysClockTo56();

#elif defined SYSCLK\_FREQ\_72MHz //我的定义的是SYSCLK\_FREQ\_72MHz，所以调用SetSysClockTo72()

  SetSysClockTo72();

#endif

}

**SetSysClockTo72()函数如下：**

static void SetSysClockTo72(void)

{

  \_\_IO uint32\_t StartUpCounter = 0, HSEStatus = 0;

   /\* SYSCLK, HCLK, PCLK2 and PCLK1 configuration ---------------------------\*/

  /\* Enable HSE \*/

  RCC->CR |= ((uint32\_t)RCC\_CR\_HSEON);

  /\* Wait till HSE is ready and if Time out is reached exit \*/

  do

  {

    HSEStatus = RCC->CR & RCC\_CR\_HSERDY;

    StartUpCounter++;

  } while((HSEStatus == 0) && (StartUpCounter != HSE\_STARTUP\_TIMEOUT));

  if ((RCC->CR & RCC\_CR\_HSERDY) != RESET)

  {

    HSEStatus = (uint32\_t)0x01;

  }

  else

  {

    HSEStatus = (uint32\_t)0x00;

  }

  if (HSEStatus == (uint32\_t)0x01)

  {

    /\* Enable Prefetch Buffer \*/

    FLASH->ACR |= FLASH\_ACR\_PRFTBE;

    /\* Flash 2 wait state \*/

    FLASH->ACR &= (uint32\_t)((uint32\_t)~FLASH\_ACR\_LATENCY);

    FLASH->ACR |= (uint32\_t)FLASH\_ACR\_LATENCY\_2;

**/\* HCLK = SYSCLK \*/**

**RCC->CFGR |= (uint32\_t)RCC\_CFGR\_HPRE\_DIV1;**

**/\* PCLK2 = HCLK \*/**

**RCC->CFGR |= (uint32\_t)RCC\_CFGR\_PPRE2\_DIV1;**

**/\* PCLK1 = HCLK \*/**

**RCC->CFGR |= (uint32\_t)RCC\_CFGR\_PPRE1\_DIV2;**

#ifdef STM32F10X\_CL

    /\* Configure PLLs ------------------------------------------------------\*/

    /\* PLL2 configuration: PLL2CLK = (HSE / 5) \* 8 = 40 MHz \*/

    /\* PREDIV1 configuration: PREDIV1CLK = PLL2 / 5 = 8 MHz \*/

**RCC->CFGR2 &= (uint32\_t)~(RCC\_CFGR2\_PREDIV2 | RCC\_CFGR2\_PLL2MUL |**

**RCC\_CFGR2\_PREDIV1 | RCC\_CFGR2\_PREDIV1SRC);**

**RCC->CFGR2 |= (uint32\_t)(RCC\_CFGR2\_PREDIV2\_DIV5 | RCC\_CFGR2\_PLL2MUL8 |**

**RCC\_CFGR2\_PREDIV1SRC\_PLL2 | RCC\_CFGR2\_PREDIV1\_DIV5);**

     /\* Enable PLL2 \*/

    RCC->CR |= RCC\_CR\_PLL2ON;

    /\* Wait till PLL2 is ready \*/

    while((RCC->CR & RCC\_CR\_PLL2RDY) == 0)

    {

    }

      /\* PLL configuration: PLLCLK = PREDIV1 \* 9 = 72 MHz \*/

**RCC->CFGR &= (uint32\_t)~(RCC\_CFGR\_PLLXTPRE | RCC\_CFGR\_PLLSRC | RCC\_CFGR\_PLLMULL);**

**RCC->CFGR |= (uint32\_t)(RCC\_CFGR\_PLLXTPRE\_PREDIV1 | RCC\_CFGR\_PLLSRC\_PREDIV1 |**

**RCC\_CFGR\_PLLMULL9);**

#else

    /\*  PLL configuration: PLLCLK = HSE \* 9 = 72 MHz \*/

    RCC->CFGR &= (uint32\_t)((uint32\_t)~(RCC\_CFGR\_PLLSRC | RCC\_CFGR\_PLLXTPRE |

                                        RCC\_CFGR\_PLLMULL));

    RCC->CFGR |= (uint32\_t)(RCC\_CFGR\_PLLSRC\_HSE | RCC\_CFGR\_PLLMULL9);

#endif /\* STM32F10X\_CL \*/

    /\* Enable PLL \*/

    RCC->CR |= RCC\_CR\_PLLON;

    /\* Wait till PLL is ready \*/

    while((RCC->CR & RCC\_CR\_PLLRDY) == 0)

    {

    }

    /\* Select PLL as system clock source \*/

**RCC->CFGR &= (uint32\_t)((uint32\_t)~(RCC\_CFGR\_SW));**

**RCC->CFGR |= (uint32\_t)RCC\_CFGR\_SW\_PLL;**

    /\* Wait till PLL is used as system clock source \*/

    while ((RCC->CFGR & (uint32\_t)RCC\_CFGR\_SWS) != (uint32\_t)0x08)

    {

    }

  }

  else

  { /\* If HSE fails to start-up, the application will have wrong clock

         configuration. User can add here some code to deal with this error \*/

  }

}

**1：AHB, APB1，APB2时钟确定**

//HCLK = SYSCLK ,从下面的分析可以得出SYSCLK是使用PLLCLK时钟的，也就是72MHZ（**至于72MHZ如何得来，请看下面分析**）

//那么就是HCLK(AHB总线时钟)=PLLCLK = 72MHZ

//AHB总线时钟等于系统时钟SYSCLK，

//**也就是 AHB时钟 = HCLK = SYSCLK = 72MHZ**

/\* HCLK = SYSCLK \*/

**RCC->CFGR |= (uint32\_t)RCC\_CFGR\_HPRE\_DIV1;**

//PLCK2等于HCLK一分频， 所以PCLK2 = HCLK，HCLK = 72MHZ,

//那么PLCK2(APB2总线时钟) = 72MHZ

//APB2总线时钟等于HCLK的一分频，也就是不分频**；**

**//APB2 时钟 = HCLK = SYSCLK = 72MHZ**

/\* PCLK2 = HCLK \*/

**RCC->CFGR |= (uint32\_t)RCC\_CFGR\_PPRE2\_DIV1;**

//PCLK1 = HCLK / 2；PCLK1 等于HCLK时钟的二分频，

//那么PCLK1(APB1) = 72MHZ / 2 = 36MHZ

//APB1总线时钟等于HCLK的二分频，也就是 **APB1时钟= HCLK / 2 = 36MHZ**

/\* PCLK1 = HCLK \*/

**RCC->CFGR |= (uint32\_t)RCC\_CFGR\_PPRE1\_DIV2;**

**2：如何得出SYSCLK（系统时钟）为72MHZ(外部晶振25MHZ)**

//记得参考英文芯片资料的时钟树P115页和RCC时钟寄存器进行理解

**RCC->CFGR2 |= (uint32\_t)(RCC\_CFGR2\_PREDIV2\_DIV5 | RCC\_CFGR2\_PLL2MUL8|RCC\_CFGR2\_PREDIV1SRC\_PLL2 | RCC\_CFGR2\_PREDIV1\_DIV5);**

**RCC\_CFGR2\_PREDIV2\_DIV5:**PREDIV2 = 5； 5分频

 也就是PREDIV2对输入的外部时钟 5分频,那么PLL2和PLL3没有倍频前

是25 /5 = 5MHZ

**RCC\_CFGR2\_PLL2MUL8  :** PLL2MUL = 8； 8倍频

 8倍频后,PLL2时钟 = 5 \* 8 = 40MHZ; 因此 PLL2CLK = 40MHZ

**RCC\_CFGR2\_PREDIV1SRC\_PLL2 :** RCC\_CFGR2的第16位为1， 选择**PLL2CLK 作为PREDIV1的时钟源**

**RCC\_CFGR2\_PREDIV1\_DIV5：PREDIV1 = 5；**PREDIV1对输入时钟5分频 **PREDIV1CLK**= PLL2CLK / 5 = 8MHZ

http://home.eeworld.com.cn/my/image/face/6.gifhttp://home.eeworld.com.cn/my/image/face/1.gif**以上是对RCC\_CFGR2进行的配置**

**--------------------------------------------------------------------------------------**

**RCC->CFGR |= (uint32\_t)(RCC\_CFGR\_PLLXTPRE\_PREDIV1 | RCC\_CFGR\_PLLSRC\_PREDIV1 |**

**RCC\_CFGR\_PLLMULL9);**

**RCC\_CFGR\_PLLXTPRE\_PREDIV1 ：**操作的是RCC\_CFGR的第17位PLLXTPRE，操作这一位和操作RCC\_CFGR2寄存器的 位[3:0]中的最低位是相同的效果

**RCC\_CFGR\_PLLSRC\_PREDIV1 ：**选择PREDIV1输出作为PLL输入时钟;PREDIV1CLK = 8MHZ,所以输入给PLL倍频的 时钟源是8MHZ

**RCC\_CFGR\_PLLMULL9 ：PLLMUL = 9；PLL倍频系数为9，也就是对 PLLCLK = PREDIV1CLK \* 8 = 72MHZ**

**以上是对RCC\_CFGR进行的配置**

---------------------------------------------------------------------------------------

//选择PLLCLK作为系统时钟源

**RCC->CFGR |= (uint32\_t)RCC\_CFGR\_SW\_PLL;**

---------------------------------------------------------------------------------------

至此基本配置已经完成，配置的时钟如下所述：

SYSCLK(系统时钟) = 72MHZ

AHB总线时钟   = 72MHZ

APB1总线时钟  = 36MHZ

APB2总线时钟  = 72MHZ

PLL时钟   = 72MHZ

PLL2时钟  = 40MHZ