AN0202

HC89F30xC/HC89F1431/ HC89F24x1/HC89F34x1C系列 MCU特色_追频功能

By Home Appliance Product Line

简介

本文主要介绍HolyChip的家电产品线（Home Appliance Product Line）的MCU产品特色，协助客户在使用产品开发过程中能够充分利用MCU的资源。

- 本技术手册适用芯片：HC89F30xC、HC89F1431、HC89F24x1、HC89F34x1C 系列芯片。
- 相关数据手册、工具、文档等等技术资料下载网址：http://www.holychip.cn/。

目录

1 应用

HolyChip 的 MCU 在出厂前已对内置高频 RC（32MHz±1%，@VDD=5V、T=25°C）进行校准，但在很多应用中，用户根据如温度、电压、时间等等外界条件变化，要求 MCU 的频率（PWM 追频、高速通信下 UART 的波特率微调、定时等）作出相应的微调。

HC89T6xx1/F1xx1 系列对时钟的微调可以分为以下两种。

一种是内部高频 RC 微调。通过这种方式进行时钟调整，可以直接调整内部高频 RC 的时钟频率。如果内部高频 RC 被用作 Fosc，则 Fosc 的频率也会跟随变化，从而影响到其他外设（UART、TIMER、PWM 等）及 CPU 的时钟。

另一种是 PWM 时钟源调整。通过这种方式进行时钟调整，可以单独调整 PWM0/3 的时钟源频率。而不会影响到 FOSC 的时钟频率，其他外设时钟及 CPU 时钟频率也不会发生变化。

HolyChip 的触摸产品中内含高频 RC 频率调整和 PWM 时钟源调整模块，同时配合 MCU 本身强大丰富的资源，非常适合雾化器、电磁炉控制等等应用。

2 内部高频RC微调

2.1 寄存器

TRMEN

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-							RCTRMEN

位编号	位符号	说明
7-1	-	保留位
0	RCTRMEN	内部高频 RC 调整使能位 1：使能内部高频 RC 调整 0：禁止内部高频 RC 调整 注：使能该寄存器后，必须立即配置 TRMV 寄存器，否则这个使能寄存器再执行完下一条指令后会被清零，内部高频 RC 调整就会失效。

TRMV

位编号	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	x	x	x	x	x	x	x
位符号	-	RCTRMV						

位编号	**位符号**	**说明**
7	-	保留位
6-0	RCTRMV	内部高频 RC 调整配置值 注： 1. x 表示不确定的值，此寄存器的上电复位值为出厂的校准值，校准后的高频 RC 时钟频率为 32MHz（1%）。 2.在配置这个寄存器值时，需要先将内部高频 RC 调整使能位配置为 1。 3.根据校准曲线软件先使能 RCTRMEN，紧接着就要配置 RCTRMV，在调整完成后 RCTRMEN 自动清零，防止重复操作。

2.2 使用注意

- 系统上电后将高频 RC 出厂校准值读出保存，该值为高频 RC=32MHz 的校准值。
- 如需微调高频 RC 频率，可按照步骤，先将 RCTRMEN 使能，再根据调节步长设置新的 RCTRMV 值。
- 使能 RCTRMEN 寄存器后必须立即配置 RCTRMV 寄存器，否则这个使能寄存器再执行完下一条指令后会被清零，内部高频 RC 调整就会失效。
- RCTRMV 的调节曲线如下图，调节步长为 0.128MHz。高频 RC 调节曲线在 RCTRMV=0x3F 时出现拐点，使用时应注意。

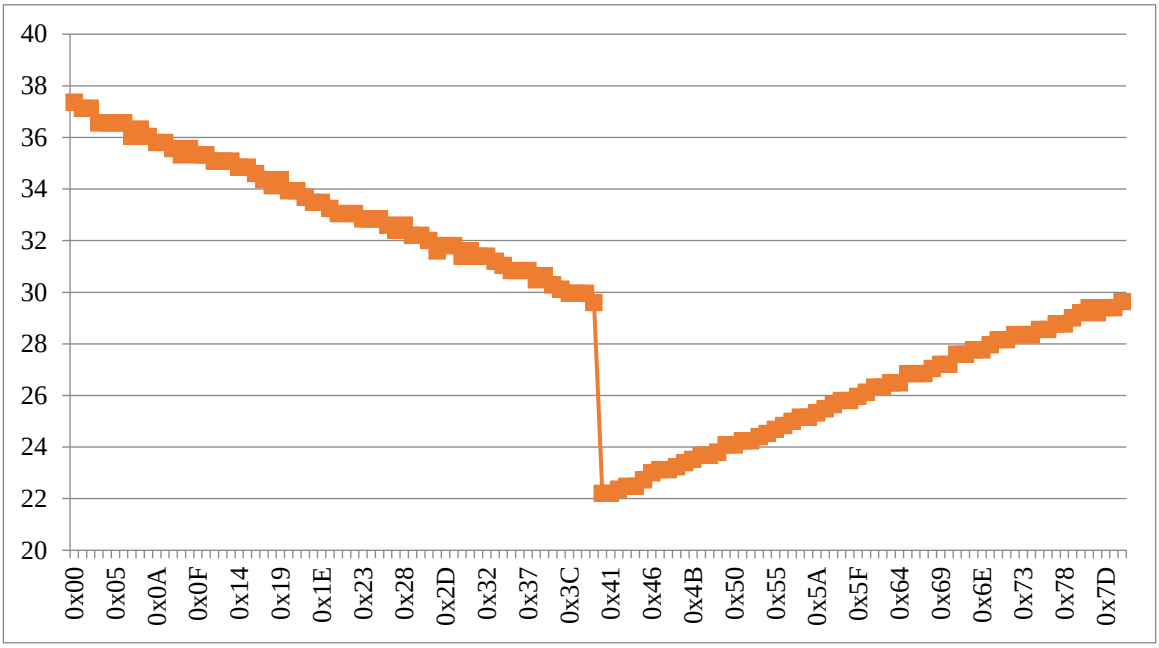


图 2-1 RCTRMV 调节曲线图

- 系统时钟选择内部高频 RC 时，重新设置 RCTRMV 值后，系统时钟频率发生改变，此时其他模块（如定时器、UART 的波特率等等）相关频率也会跟随改变。

2.3 精度

使用内部高频 RC 微调模块，用户在计算时应参考频率校准曲线图，调节步长为 0.128MHz，则调整精度为

$$调整精度=\frac{0.128MHz}{32MHz}*100\%=0.4\%$$

2.4 典型应用举例

HolyChip 的触摸资料中提供了一个关于内部高频 RC 调整的例程（“CLK-内部高频 RC 调整”），实现效果为：微调降低内部高频 RC 频率，从而减少串口 115200 波特率的误差。

代码如下：

```
#define       ALLOCATE_EXTERN
#include " HC89F30xC.H"

#define TrmvAdjustedValue 5                                    //计算后需要调整的 TRMV 值
void TRMVAdjust(unsigned char fuc_AdjustValue);      //频率调整函数

/*******************************************************************************************
  * @实现效果    微调降低内部高频 RC 频率，从而减少串口 115200 波特率的误差
*******************************************************************************************/
void main()
{
/************************************系统初始化*************************************/
    CLKSWR = 0x50;                                     //选择内部高频 RC 为系统时钟，内部高频 RC 1 分频，Fosc=32MHz
    CLKDIV = 0x02;                                     //Fosc 2 分频得到 Fcpu，Fcpu=16MHz
/*********************************频率初始化调整*************************************/
    //时钟为 32MHz 时波特率计算
    //波特率  = 1/16 * (T4 时钟源频率  /  定时器 4 预分频比) / (65536 - 0xFFEF)
    //          = 1/16 * ((32000000 / 1) / 17)
    //              = 117647.06(误差 2.12%)

    //反推最佳系统时钟
    //T4 时钟源频率  = 115200 / (1/16) * (65536 - 0xFFEF)
    //                        = 115200 * 16 * 17
    //                        = 31334400Hz

    //相差频率及 RCTRMV 值计算
    //RCTRMV = (32000000 - 31334400) / 128000
    //              = 5.2  ≈  5

    //调整后频率计算
    //调整后频率  = 32000000 - 5 * 128000
    //                      = 31360000

    //调整后波特率计算
    //波特率  = 1/16 * (T4 时钟源频率  /  定时器 4 预分频比) / (65536 - 0xFFEF)
    //              = 1/16 * ((31360000 / 1) / 17)
    //              = 115294.12(误差 0.08%)

    TRMVAdjust(TrmvAdjustedValue);          //频率调整
```

```
    P0M3 = 0xC1;                        //P03 设置为推挽输出
    CLKO_MAP = 0x03;                    //时钟输出，映射 P03
    CLKOUT = 0x17;                      //使能内部高频时钟 8 分频输出
                                        //输出频率为 3920000Hz

    P0M6 = 0xC1;                        //P06 设置为推挽输出
    P0M7 = 0x61;                        //P07 设置为上拉输出
    TXD_MAP = 0x06;                     //TXD 映射 P06
    RXD_MAP = 0x07;                     //RXD 映射 P07
    T4CON = 0x06;                       //T4 工作模式：UART1 波特率发生器
    TH4 = 0xFF;
    TL4 = 0xEF;                         //波特率 115200
    SCON2 = 0x02;                       //8 位 UART，波特率可变
    SCON = 0x10;                        //允许串行接收
    IE |= 0x10;                         //使能串口中断
    EA = 1;                             //使能总中断
    while(1)
    {
        SBUF = 0x55;//发送 8 位串口数据
        while(!(SCON & 0x02));
        SCON &=~ 0x02;
    }
}

/**
  * @说明        频率调整
  * @参数        fuc_AdjustValue ：需要调整的值
  * @返回值 无
  */
#pragma disable                         //确保调整时不会进中断导致调整失败
void TRMVAdjust(unsigned char fuc_AdjustValue)
{
    unsigned char fuc_TrmvValue = 0;        //TRMV 的值
    unsigned char fuc_TrmvCalcValue = 0;    //计算中转值

    fuc_TrmvValue = TRMV;                   //读取 TRMV 的值
    if(fuc_TrmvValue >= (0x3F + fuc_AdjustValue))
    {
        fuc_TrmvValue -= fuc_AdjustValue;
    }
    else if((fuc_TrmvValue >= 0x3F) && (fuc_TrmvValue < (0x3F + fuc_AdjustValue)))
    {
        fuc_TrmvValue = 0x3F;
    }
    else if(fuc_TrmvValue <= (0x3F - fuc_AdjustValue))
    {
        fuc_TrmvValue += fuc_AdjustValue;
```

```
    }
    else if((fuc_TrmvValue > (0x3F - fuc_AdjustValue)) && (fuc_TrmvValue < 0x3F))
    {
        fuc_TrmvCalcValue = 0x3F - fuc_TrmvValue;
        fuc_TrmvCalcValue = fuc_AdjustValue - fuc_TrmvCalcValue;
        fuc_TrmvValue = 0x7F - fuc_TrmvCalcValue;
    }
    TRMEN |= 0x01;                          //使能内部高频 RC 调整
    TRMV = fuc_TrmvValue;                   //写入校准后的值
}
```

3 PWM时钟源调整

3.1 寄存器

SFTR

位编号	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R/W	R/W
复位值	0	0	0	0	0	0	0	0
位符号	-						SFTREN1	SFTREN0

位编号	位符号	说明
7-2	-	保留位
1	SFTREN1	PWM0 时钟源软件调整使能位 1：允许软件单独调整 PWM0 时钟源，不影响其他模块时钟 0：使用芯片出厂校准后的标准 32MHz RC 作为 PWM 时钟源
0	SFTREN0	PWM3 时钟源软件调整使能位 1：允许软件单独调整 PWM3 时钟源，不影响其他模块时钟 0：使用芯片出厂校准后的标准 32MHz RC 作为 PWM 时钟源 注意：SFTREN1 和 SFTREN0 不允许同时为 1

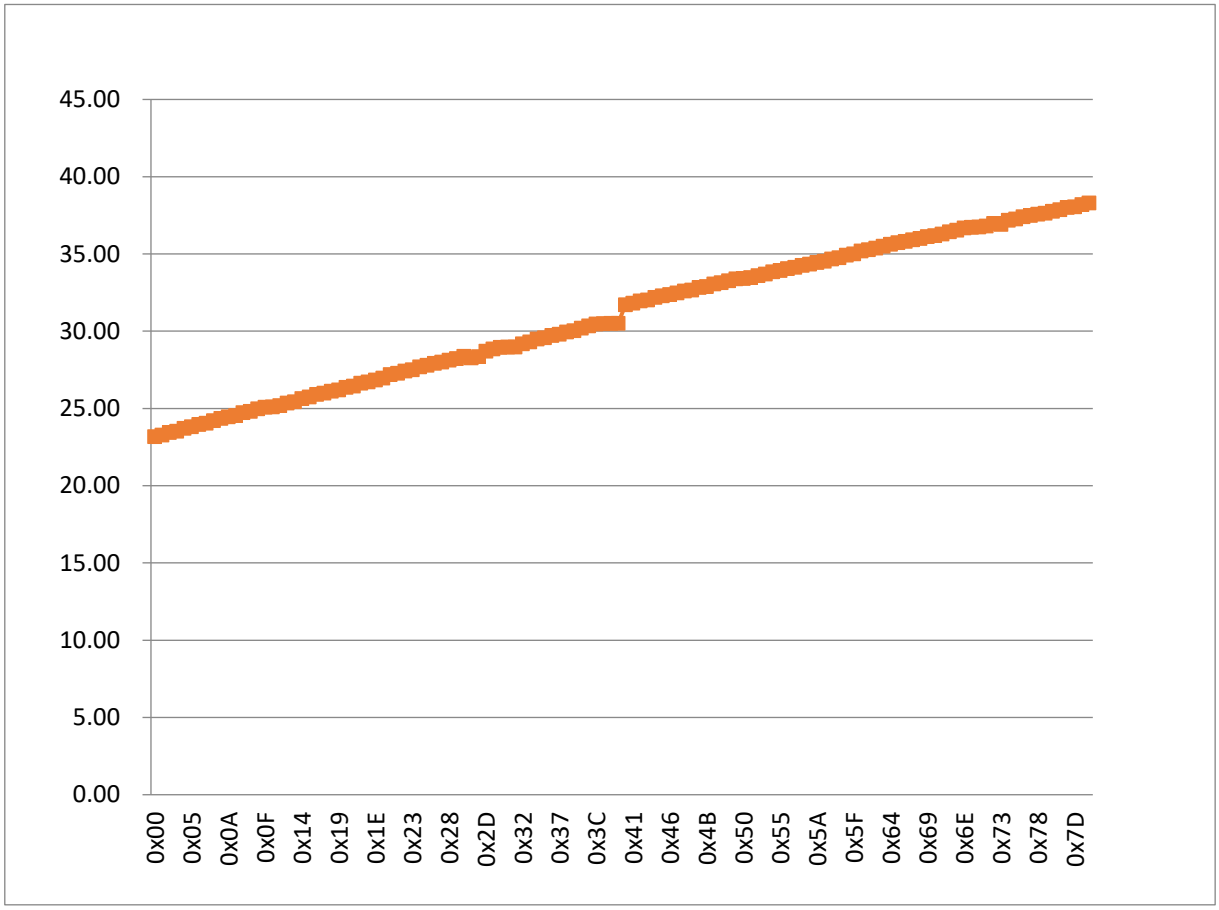
SFTRMV

位编号	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位值	0	x	x	x	x	x	x	x
位符号	-	SFTRMV						

位编号	位符号	说明
7	-	保留位
6-0	SFTRMV	PWM0/3 时钟源调整配置值 注： 1. x 表示不确定的值，此寄存器的上电复位值为出厂的校准值，校准后的高频 RC 时钟频率为 32MHz（1%）。 2.在配置这个寄存器值时，需要先将 SFTRENx（x=0、1）使能位配置为 1。

3.2 使用注意

- 使能 PWM 时钟源调整后，PWM 的时钟源会由 Fosc 时钟切换成 PWM32MHz 时钟。且仅 PWM 时钟频率发生改变。
- 如需调整 PWM 时钟源，可按照步骤，先将 SFTRENx 的使能，再根据调节步长设置新的 SFTRMV 值。
- SFTRMV 的调节曲线如下图，调节步长为 0.128MHz。PWM 时钟源调整曲线在 SFTRMV=0x40 时出现突变，使用时应注意。



3.3 精度

使用 PWM 时钟源调整模块，用户在计算时应参考频率校准曲线图，调节步长为 0.128MHz，则调整精度为

$$调整精度=\frac{0.128MHz}{32MHz}*100\%=0.4\%$$

3.4 经典应用举例

HolyChip 的触摸资料中提供了一个关于 PWM 时钟源调整的例程（“CLK-PWM 时钟源调整”），实现效果为：P03 口循环输出 15.69khz、31.87khz 的方波，每种频率持续输出 500ms。

代码如下：

```
#define       ALLOCATE_EXTERN
#include " HC89F30xC.H"
void Delay_ms(unsigned int fui_i);
/*******************************************************************************************
  * @实现效果            P03 口循环输出 15.69khz、31.87khz 的方波，每种频率持续 500ms
*******************************************************************************************/
void main()
{
/*************************************系统初始化*************************************/
      CLKSWR = 0x51;                                  //选择内部高频 RC 为系统时钟，内部高频 RC 1 分频，Fosc=16MHz
      CLKDIV = 0x01;                                  //Fosc 1 分频得到 Fcpu，Fcpu=16MHz
/*************************************频率初始化调整*************************************/
    P0M3 = 0xC1;                                  //P03 设置为推挽输出
      PWM3_MAP = 0x03;                                //PWM3 映射 P03 口
      //周期计算  = 0xFF / (Fosc / PWM 分频系数)            （Fosc 见系统时钟配置的部分）
      //                  = 0xFF /(16000000 / 4)
      //                  = 255 /4000000
      //                  = 63.75us           即 15.69KHZ

      PWM3P = 0xFF;                                   //PWM 周期为 0xFF
      //有效电平时间计算（即占空比）
      //                  = 0x55 / (Fosc / PWM 分频系数)              （Fosc 见系统时钟配置的部分）
      //                  = 0x55 /(16000000 / 4)
      //                  = 85 /4000000
      //                  = 21.25us           占空比为 21.25 / 63.75 = 34%

      PWM3D = 0x55;                                   //PWM 占空比设置
      PWM3C = 0x92;                                   //使能 PWM3，关闭中断，允许输出，时钟 4 分频
      while(1)
      {
          Delay_ms(500);
          SFTR =1 ;                   //PWM32M  软件调整使能
          SFTRMV = SFTRMV;          //PWM32M  调整
          Delay_ms(500);
          SFTR =0 ;                    //PWM32M  软件调整失能
      }
}
/**
  * @说明          延时函数
```

```
   * @参数          fui_i：延时时间
   * @返回值  无
   * @注     Fcpu = 16MHz，fui_i = 1 时，延时时间约为 1Ms
   */
void Delay_ms(unsigned int fui_i)
{
        unsigned int fui_j;
        for(;fui_i > 0;fui_i --)
        for(fui_j = 1596;fui_j > 0;fui_j --);
}
```

4 PWM频率高精度步进调整实现方法

HolyChip 的 HC89F30xC 系列芯片通过修改周期寄存器[PWMxPH:PWMxPL]无法对 PWM 进行较高精度的调整。

HC89F30xC 系列芯片提供了两种方法实现 PWM 高精度调整，第一种方法是改变内部高频 RC 时钟频率来实现 PWM 高精度调整，第二种方法是改变 PWM 的时钟源频率来实现 PWM 高精度调整。第二种方法与第一种方法类似，下面将以第一种方法进行解说。

通过改变内部高频 RC 时钟频率来实现 PWM 高精度调整，步进精度约 0.4%。用户可以通过使能 TRMEN 寄存器、修改 TRMV 寄存器（可参考本文第 2 章节内部高频 RC 微调寄存器）里的值改变内部高频 RC 频率（需要将内部高频 RC 时钟设置为 Fosc，**此时所有使用 Fosc 作时钟源的模块的频率都会发生变化**），从而实现修改 PWM 频率。

内部高频 RC 频率与 PWM 频率及 TRMV[6:0]关系如下表所示:

序号	TRMV[6:0]值	内部高频RC频率 (kHz)	[PWM0PH: PWM0PL]	PWM频率 (kHz)
1	TRMV[x]+n	32000-128*n	P	（32000-128*n）/（P+1）
2	...	...	P	...
3	TRMV[x]+3	32000-128*3=31616	P	（32000-128*3）/（P+1）
4	TRMV[x]+2	32000-128*2=31744	P	（32000-128*2）/（P+1）
5	TRMV[x]+1	32000-128*1=31872	P	（32000-128*1）/（P+1）
6	TRMV[x]	32000	P	32000/（P+1）
7	TRMV[x]-1	32000+128*1=32128	P	（32000-128*1）/（P+1）
8	TRMV[x]-2	32000+128*2=32256	P	（32000-128*2）/（P+1）
9	TRMV[x]-3	32000+128*3=32384	P	（32000-128*3）/（P+1）
10	...	...	P	...
11	TRMV[x]-n	32000+128*n	P	（32000-128*n）/（P+1）

表 4-1 内部高频 RC 频率与 PWM 频率及 TRMV[6:0]关系

下面以常用的 1.7M/2.4M/3MHz 几个 PWM 频率为例，下表说明 TRMV[6:0]值与内部高频 RC 频率和 PWM 频率的关系。

参数\PWM频率	1.7MHz±1%	2.4MHz±1%	3MHz±1%	
周期最佳值 [PWM0PH:PWM0PL]	18	12	9	10
内部高频 RC 工作频率范围（MHz）	32.3±1%（31.977~32.623）	31.2±1%（30.888~31.512）	30±1%（29.7~30.3，次优）	33±1%（32.67~33.33，最优）
TRMV 偏移量 （TRMV[6:0]-TRMV[x]）	1~5	-9~-3	-18~-13	5~11
TRMV[6:0]值有效范围	0~127	0~127	0~127	0~127

表 4-2

下表以[PWM0PH:PWM0PL]=10 为例说明 PWM 频率为 3MHz 时，PWM 频率的步进输出频率和参数的关系，可以看出 PWM 频率在 3MHz 的±1%的范围内可以实现 11.6kHz 左右的步进，步进调整精度约为 11.6/3000=0.39%。

TRMV 偏移量 （TRMV[6:0]-TRMV[x]）	0	-1	…	-6	-7	-8	-9	…
内部高频 RC 工作频率范围 （KHz）	32000	32128	...	32768	32896	33024	33152	...
PWM 周期 （[PWM0PH:PWM0PL]+1）	11	11	...	11	11	11	11	...
PWM 频率 （KHz）	2909.1	2920.7	...	2978.9	2990.2	3002.2	3013.8	...

表 4-3 PWM 频率为 3MHz 时 TRMV 时偏移量分析

用户如需得到其他频率的 PWM，可以先确定最接近所需的 PWM 频率时的[PWM0PH: PWM0PL]的值，再进行 TRMV[6:0]值修改，从而实现 PWM 高精度的调整。

注：

- Fosc=32MHz，PWM 时钟源=Fosc/1。
- 内部高频 RC 出厂时设定的时 32MHz，时钟微调步进 0.128MHz，每颗 IC 会略有偏差。
- **需注意 TRMV[6:0]值在 0x3F 的时候内部高频 RC 会有一个拐点**（请参考图 2-1 RCTRMV 的调节曲线）。
- **以上例子仅以 TRMV[x]+偏移量的值和 TRMV[x]值在调节曲线的拐点（0x3F）左侧进行解说。其他情况的计算与此类似，仅需注意拐点位置和内部高频 RC 的频率的变化趋势的即可。**

5 版本说明

版本	日期	描述
V1.00	2020/07/30	初版
V1.01	2021/06/02	HC88T6xx1 修改替换成 HC89F30xC
V1.02	2021/11/03	删除 HC88F1xx1 系列
V1.03	2024/11/13	增加 HC89F1431、HC89F24x1、HC89F34x1C 系列

芯圣电子

2024 年 11 月