

FM33LG0xxH 替换 FM33LG0XX 软件说明

概述

FM33LG0XXH以兼容FM33LG0XX为设计目标，但由于部分IP升级，有些外设仍然有些小的变更。为方便用户在旧项目上替换，本文档描述使用FM33LG0XX的驱动怎样进行简单修改就可以在FM33LG0XXH上直接运行。
新开发项目建议使用FM33LG0XXH的驱动进行开发。

差异说明

ADC

	FM33LG0xxH	FM33LG0XX
ADC 时钟	使用 APBCLK	RCLF\RCHF\XTHF\PLL 四选1

说明：
FM33LG0XXH的ADC工作时钟为APBCLK，FM33LG0XX的ADC工作时钟可以在RCLF\RCHF\XTHF\PLL中四选一。
但FM33LG0xx程序系统时钟选择PLL，ADC工作时钟非PLL时，使用FM33LG0XXH芯片实际工作时钟是APBCLK的6分频，需要按照实际的PLL时钟评估采样时间是否足够。

举例：
FM33LG0XX系统时钟是RCHF，ADC的工作时钟选择RCHF，使用FM33LG0XXH芯片时APBCLK就是RCHF软件不用修改。
FM33LG0XX系统时钟是PLL32M，ADC的工作时钟选择RCHF8M，使用FM33LG0XXH芯片实际工作时钟是APBCLK32M的6分频5.3M，采样时间变长不影响采样精度，软件可以不用修改。

特殊说明

FM33LG0XXH的ADC在采样时有万分之几的概率出现采样值偏高，建议ADC采样时多采样几次，扔掉最小值和最大值

PLL

	FM33LG0xxH	FM33LG0XX
参考时钟	输入参考时钟4MHz	输入参考时钟1MHz

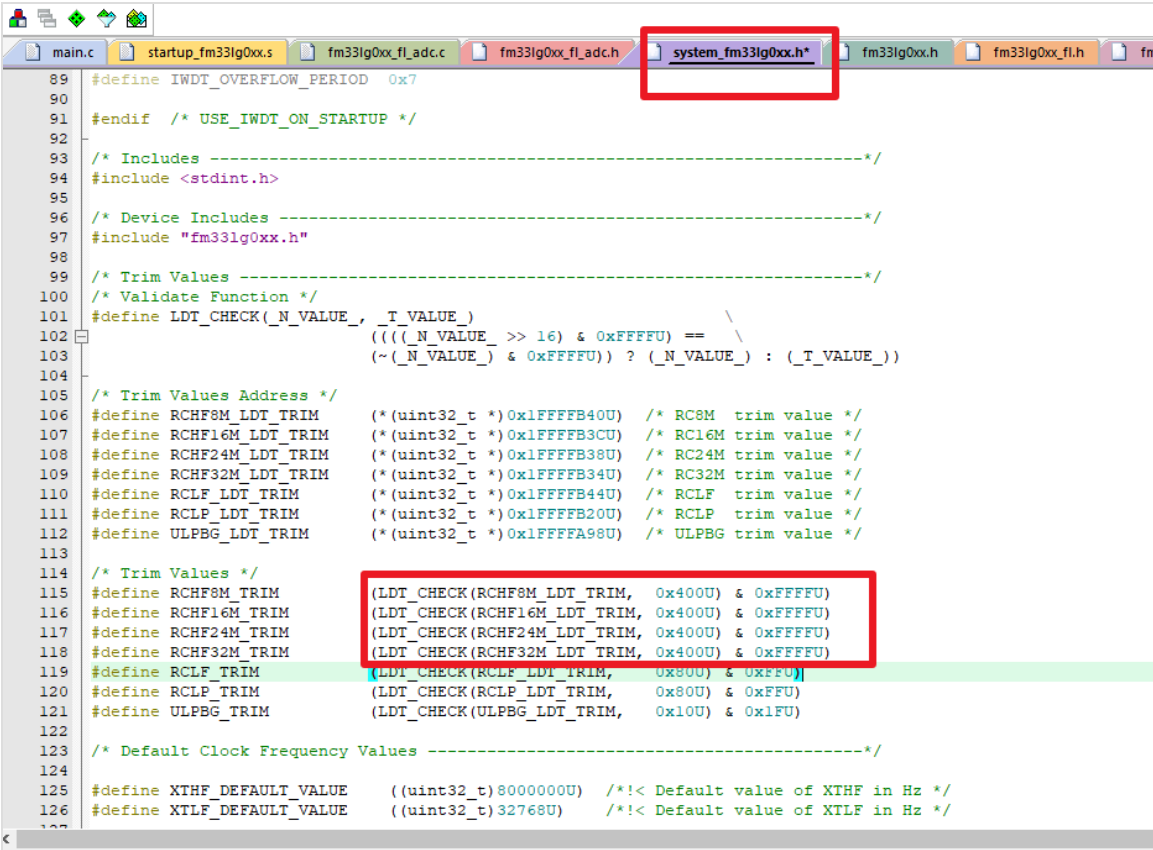
PLL范围	32M~48M	32M~64M
-------	---------	---------

说明：
FM33LG0xxH的PLL时钟源需要分频到4M，再进行倍频，倍频的范围32M~48M
FM33LG0xx的PLL时钟源需要分频到1M，再进行倍频，倍频的范围32M~64M
举例： FM33LG0xxH的PLL时钟源为RCHF8M，需要先2分频到4M，再倍频到32M~48M

RCHF

	FM33LG0xxH	FM33LG0XX
TRIM	11bit	8bit

说明：
FM33LG0XXH的TRIM值是11bit,需要修改驱动 `system_fm33lg0xx.h` 文件,按下图修改。



```
89 #define IWDI_OVERFLOW_PERIOD 0x7
90
91 #endif /* USE_IWDI_ON_STARTUP */
92
93 /* Includes -----*/
94 #include <stdint.h>
95
96 /* Device Includes -----*/
97 #include "fm33lg0xx.h"
98
99 /* Trim Values -----*/
100 /* Validate Function */
101 #define LDT_CHECK(_N_VALUE_, _T_VALUE_) \
102     ((((_N_VALUE_ >> 16) & 0xFFFFU) == \
103      (~(_N_VALUE_) & 0xFFFFU)) ? (_N_VALUE_) : (_T_VALUE_))
104
105 /* Trim Values Address */
106 #define RCHF8M_LDT_TRIM      (*(uint32_t *)0x1FFFFB40U) /* RC8M trim value */
107 #define RCHF16M_LDT_TRIM    (*(uint32_t *)0x1FFFFB3CU) /* RC16M trim value */
108 #define RCHF24M_LDT_TRIM    (*(uint32_t *)0x1FFFFB38U) /* RC24M trim value */
109 #define RCHF32M_LDT_TRIM    (*(uint32_t *)0x1FFFFB34U) /* RC32M trim value */
110 #define RCLF_LDT_TRIM       (*(uint32_t *)0x1FFFFB44U) /* RCLF trim value */
111 #define RCLP_LDT_TRIM       (*(uint32_t *)0x1FFFFB20U) /* RCLP trim value */
112 #define ULPBG_LDT_TRIM      (*(uint32_t *)0x1FFFA98U) /* ULPBG trim value */
113
114 /* Trim Values */
115 #define RCHF8M_TRIM          (LDT_CHECK(RCHF8M_LDT_TRIM, 0x400U) & 0xFFFFU)
116 #define RCHF16M_TRIM        (LDT_CHECK(RCHF16M_LDT_TRIM, 0x400U) & 0xFFFFU)
117 #define RCHF24M_TRIM        (LDT_CHECK(RCHF24M_LDT_TRIM, 0x400U) & 0xFFFFU)
118 #define RCHF32M_TRIM        (LDT_CHECK(RCHF32M_LDT_TRIM, 0x400U) & 0xFFFFU)
119 #define RCLF_TRIM           (LDT_CHECK(RCLF_LDT_TRIM, 0x80U) & 0xFFU)
120 #define RCLP_TRIM           (LDT_CHECK(RCLP_LDT_TRIM, 0x80U) & 0xFFU)
121 #define ULPBG_TRIM          (LDT_CHECK(ULPBG_LDT_TRIM, 0x10U) & 0x1FU)
122
123 /* Default Clock Frequency Values -----*/
124
125 #define XTHF_DEFAULT_VALUE   ((uint32_t)8000000U) /*!< Default value of XTHF in Hz */
126 #define XTILF_DEFAULT_VALUE  ((uint32_t)32768U) /*!< Default value of XTILF in Hz */
```

adc.cmain.csystem_fm33lg0xx.cfm33lg0xx_fl_vref.hfm33lg0xx_fl_adc.hfm33lg0xx_fl_rmu.hfm33lg0xx_fl_cmu.h*

```
118 #define CMU_RCHFCR_FSEL CMU_RCHFCR_FSEL_Msk
119
120 #define CMU_RCHFCR_EN_Pos (0U)
121 #define CMU_RCHFCR_EN_Msk (0x1U << CMU_RCHFCR_EN_Pos)
122 #define CMU_RCHFCR_EN CMU_RCHFCR_EN_Msk
123
124 #define CMU_RCHFTR_TRIM_Pos (0U)
125 #define CMU_RCHFTR_TRIM_Msk (0x7ffU << CMU_RCHFTR_TRIM_Pos)
126 #define CMU_RCHFTR_TRIM CMU_RCHFTR_TRIM_Msk
127
128 #define CMU_PLLCR_EN_Pos (0U)
129 #define CMU_PLLCR_EN_Msk (0x1U << CMU_PLLCR_EN_Pos)
130 #define CMU_PLLCR_EN CMU_PLLCR_EN_Msk
```

adc.cmain.csystem_fm33lg0xx.cfm33lg0xx_fl_vref.hfm33lg0xx_fl_adc.hfm33lg0xx_fl_rmu.hfm33lg0xx_fl_cmu.h

```
887 /**
888  * @brief Set RCHF Frequency Trim Value
889  * @rmtoll RCHFTR TRIM FL_CMU_RCHF_WriteTrimValue
890  * @param value TrimValue The value of RCHF trim
891  * @retval None
892  */
893 STATIC_INLINE void FL_CMU_RCHF_WriteTrimValue(uint32_t value)
894 {
895     MODIFY_REG(CMU->RCHFTR, (0x7ffU << 0U), (value << 0U));
896 }
897
898 /**
899  * @brief Get RCHF Frequency Trim Value
900  * @rmtoll RCHFTR TRIM FL_CMU_RCHF_ReadTrimValue
901  * @retval The value of RCHF trim
902  */
903 STATIC_INLINE uint32_t FL_CMU_RCHF_ReadTrimValue(void)
904 {
905     return (uint32_t)(READ_BIT(CMU->RCHFTR, (0x7ffU << 0U)) >> 0U);
906 }
907
908 /**
```

FLASH

	FM33LG0xxH	FM33LG0XX
预取指	默认使能	默认关闭
wait	16M 1wait	24M 1wait

说明:
使用FM33LG0XXH芯片在复位初始化时关闭预取指，指令：FLASH -> PFCR = 0。
FM33LG0xxH 系统时钟小于等于16M不插入wait，大于16M小于等于32M插入1wait。
FM33LG0xx系统时钟小于等于24M不插入wait，大于24M小于等于48M插入1wait。
举例:
FM33LG0XX系统时钟是RCHF8M，RCHF16M，使用FM33LG0XXH芯片时程序不用修改。
FM33LG0XX系统时钟是PLL32M，使用FM33LG0XXH芯片时程序不用修改。

VREF

	FM33LG0xxH	FM33LG0XX
电压	1V	1.22V
建立时间	忽略	1.5ms
功耗	忽略	1.5ua

中断	无	建立中断
----	---	------

说明：
VREF用于ADC采样时反推VDD电压值，FM33LG0XXH和FM33LG0XX关于VREF的ADC采样、ADC计算公式完全兼容不需要修改。
FM33LG0XX的VREF1P2用户在休眠前后可能会有关闭和使能操作，同样不需要修改程序可以在FM33LG0XXH上兼容运行。
FM33LG0XXH的VREF相对FM33LG0XX的VREF1P2删除了VREF的建立中断，如用到需修改程序。

DAC

说明：
FM33LG0XXH删除了采样保持模式。
FM33LG0XXH DAC使能后默认输出VDD，使用时需要注意。FM33LG0XX可以在DAC使能前写数据寄存器，使能后输出该电平。

COMP

说明：
FM33LG0XXH相比FM33LG0XX删除比较器3，且比较器0和1的管脚分布两颗芯片不完全兼容，如用到需参看FM33LG0XXH的封装图进行修改。

CAN

说明： FM33LG0XXH的CAN模块做了升级，使用时需要按照FM33LG0XXH的例程进行修改。

VREP

说明： FM33LG0XXH已经删除VREFP模块

休眠

说明： FM33LG0xxH休眠唤醒时多了LDO的建立时间寄存器T0A，程序不需要配置。

FOUT

说明： FM33LG0xxH的FOUT时钟源列表有变更，FOUT一般用于调试，程序没有用到可以不用关注。

OPA

说明： FM33LG0xxH删除了OPA模块。

RCLF

说明：FM33LG0xxH删除了RCLF模块。

UART2

说明：FM33LG0xxH新增UART2模块，UART模块有LIN功能。

版本

版本	日期	说明
V1.0	2026-08-12	初始版本
V1.1	2026-08-18	修改FLASH预取值处理、修正VREF说明
V1.2	2026-09-09	增加CAN的说明
V1.3	2026-10-16	增加ADC说明