

STM32 固件库中文使用手册只到 V2.0，其后未再更新。本文就 V3.5 对 V2.0 的常用部件变更做简要说明。

## ADC\_

无不同

## BKP\_

无不同

## DMA\_

void DMA\_SetCurrDataCounter(DMA\_Channel\_TypeDef\* DMAy\_Channelx, uint16\_t DataNumber); V3.5 多出  
向通道 x 的 CNDTR 寄存器写入数据传输数量

## EXTI\_

无不同

## GPIO\_

void GPIO\_ETH\_MediaInterfaceConfig(uint32\_t GPIO\_ETH\_MediaInterface); V3.5 多出  
选用以太网媒体界面

## IWDG\_

无不同

## PWR\_

无不同

## RCC\_

V2.0 的所有函数均有，另增加以下函数：

```
#if defined (STM32F10X_LD_VL) || defined (STM32F10X_MD_VL) || defined  
(STM32F10X_HD_VL) || defined (STM32F10X_CL)  
void RCC_PREDIV1Config(uint32_t RCC_PREDIV1_Source, uint32_t RCC_PREDIV1_Div);  
#endif
```

```
#ifdef STM32F10X_CL  
void RCC_PREDIV2Config(uint32_t RCC_PREDIV2_Div);  
void RCC_PLL2Config(uint32_t RCC_PLL2Mul);  
void RCC_PLL2Cmd(FunctionalState NewState);  
void RCC_PLL3Config(uint32_t RCC_PLL3Mul);  
void RCC_PLL3Cmd(FunctionalState NewState);  
#endif /* STM32F10X_CL */
```

```
#ifndef STM32F10X_CL
```

```

#else
    void RCC_OTGFSCLKConfig(uint32_t RCC_OTGFSCLKSource);
#endif /* STM32F10X_CL */

#ifdef STM32F10X_CL
    void RCC_I2S2CLKConfig(uint32_t RCC_I2S2CLKSource);
    void RCC_I2S3CLKConfig(uint32_t RCC_I2S3CLKSource);
#endif /* STM32F10X_CL */

#ifdef STM32F10X_CL
    void RCC_AHBPeriphResetCmd(uint32_t RCC_AHBPeriph, FunctionalState NewState);
#endif /* STM32F10X_CL */

```

## RTC\_

无不同

## SPI\_

无不同

## TIM\_

V3.5 取消了 TIM1，将其并入 TIMx（x=1），若干结构变量做了相应的扩展。使用时 x≠1 时按照原 TIMx 设置，x=1 时按原 TIMx 设置。另外以下函数有变化：

```

void TIM_CCxCmd(TIM_TypeDef* TIMx, uint16_t TIM_Channel, uint16_t TIM_CCx); V3.5
void TIM_CCxNCmd(TIM_TypeDef* TIMx, uint16_t TIM_Channel, uint16_t TIM_CCxN); V3.5
void TIM1_CCxCmd(u16 TIM1_Channel, FunctionalState Newstate); V2.0
void TIM1_CCxNCmd(u16 TIM1_Channel, FunctionalState Newstate); V2.0

```

功能：使能或失能捕获比较通道 x 和 xN  
 参数 Newstate(DISABLE 或 ENABLE)改成 TIM\_CCx(TIM\_CCx\_Enable 或 TIM\_CCx\_Disable)  
 和 TIM\_CCxN(TIM\_CCxN\_Enable 或 TIM\_CCxN\_Disable)

```

void TIM_SetIC1Prescaler(TIM_TypeDef* TIMx, uint16_t TIM_ICPSC); V3.5
void TIM_SetIC2Prescaler(TIM_TypeDef* TIMx, uint16_t TIM_ICPSC); V3.5
void TIM_SetIC3Prescaler(TIM_TypeDef* TIMx, uint16_t TIM_ICPSC); V3.5
void TIM_SetIC4Prescaler(TIM_TypeDef* TIMx, uint16_t TIM_ICPSC); V3.5
void TIM1_SetIC1Prescaler(u16 TIM1_IC1Prescaler); V2.0
void TIM1_SetIC2Prescaler(u16 TIM1_IC2Prescaler); V2.0
void TIM1_SetIC3Prescaler(u16 TIM1_IC3Prescaler); V2.0
void TIM1_SetIC4Prescaler(u16 TIM1_IC4Prescaler); V2.0

```

功能：设置输入捕获预分频值（TIM\_ICPSC\_DIV1、TIM\_ICPSC\_DIV2、TIM\_ICPSC\_DIV4、TIM\_ICPSC\_DIV8），分别为输入每 1、2、4 和 8 个边沿触发一次捕获

## USART\_

```

void USART_OverSampling8Cmd(USART_TypeDef* USARTx, FunctionalState NewState); V3.5

```

多出

打开或关闭 USARTx 对单频正弦信号的模式

USARTx: USART1, USART2, USART3, UART4 or UART5.

NewState: ENABLE 或 DISABLE

void USART\_OneBitMethodCmd(USART\_TypeDef\* USARTx, FunctionalState NewState); V3.5

多出

打开或关闭 USARTx 一点的抽样方法

USARTx: USART1, USART2, USART3, UART4 or UART5.

NewState: ENABLE 或 DISABLE

## WWDG\_

无不同

## NVIC\_

对 NVIC 的操作，仅保留以下 4 个函数：

void NVIC\_PriorityGroupConfig(uint32\_t NVIC\_PriorityGroup);

void NVIC\_Init(NVIC\_InitTypeDef\* NVIC\_InitStruct);

void NVIC\_SetVectorTable(uint32\_t NVIC\_VectTab, uint32\_t Offset);

void NVIC\_SystemLPConfig(uint8\_t LowPowerMode, FunctionalState NewState);

## SysTick\_

对 SysTick 的操作，仅保留以下 1 个函数：

void SysTick\_CLKSourceConfig(uint32\_t SysTick\_CLKSource);

功能：配置 SysTick 时钟源

参数：

SysTick\_CLKSource:

SysTick\_CLKSource\_HCLK\_Div8:根据时钟除以8选为SysTick时钟源。

SysTick\_CLKSource\_HCLK:根据SysTick时钟选为时钟源。

uint32\_t SysTick\_Config(uint32\_t ticks);

功能：配置 SysTick 重装计数值，使能中断，启动运行

参数：

ticks: 24 位以内的重装值

返回：1失败,0成功