Stm32 寄存器与库函数概览（摘自固件库使用手册）

Table 1. 本文档所有缩写定义

缩写	外设/单元
ADC	模数转换器
BKP	备份寄存器
CAN	控制器局域网模块
DMA	直接内存存取控制器
EXTI	外部中断事件控制器
FLASH	闪存存储器
GPIO	通用输入输出
I2C	内部集成电路
IWDG	独立看门狗
NVIC	嵌套中断向量列表控制器
PWR	电源/功耗控制
RCC	复位与时钟控制器
RTC	实时时钟
SPI	串行外设接口
SysTick	系统嘀嗒定时器
TIM	通用定时器
TIM1	高级控制定时器
USART	通用同步异步接收发射端
WWDG	窗口看门狗

4. 模拟/数字转换器

模拟/数字转换器（ADC）是一种提供可选择多通道输入，逐次逼近型的模数转换器。分辨率为 12 位。

Table 4. ADC 寄存器

寄存器	描述
SR	ADC 状态寄存器
CR1	ADC 控制寄存器 1
CR2	ADC 控制寄存器 2
SMPR1	ADC 采样时间寄存器 1
SMPR2	ADC 采样时间寄存器 2
JOFR1	ADC 注入通道偏移寄存器 1
JOFR2	ADC 注入通道偏移寄存器 2
JOFR3	ADC 注入通道偏移寄存器 3
JOFR4	ADC 注入通道偏移寄存器 4
HTR	ADC 看门狗高阈值寄存器
LTR	ADC 看门狗低阈值寄存器
SQR1	ADC 规则序列寄存器 1
SQR2	ADC 规则序列寄存器 2
SQR3	ADC 规则序列寄存器 3
JSQR1	ADC 注入序列寄存器
DR1	ADC 规则数据寄存器 1

DR2	ADC 规则数据寄存器 2
DR3	ADC 规则数据寄存器 3
DR4	ADC 规则数据寄存器 4

Table 5. ADC 固件库函数

函数名	描述
ADC_DeInit	将外设 ADCx 的全部寄存器重设为缺省值
ADC_Init	根据 ADC_InitStruct 中指定的参数初始化外设 ADCx 的寄存器
ADC_StructInit	把 ADC_InitStruct 中的每一个参数按缺省值填入
ADC_Cmd	使能或者失能指定的 ADC
ADC_DMACmd	使能或者失能指定的 ADC 的 DMA 请求
ADC_ITConfig	使能或者失能指定的 ADC 的中断
ADC_ResetCalibration	重置指定的 ADC 的校准寄存器
ADC_GetResetCalibrationStatus	获取 ADC 重置校准寄存器的状态
ADC_StartCalibration	开始指定 ADC 的校准程序
ADC_GetCalibrationStatus	获取指定 ADC 的校准状态
ADC_SoftwareStartConvCmd	使能或者失能指定的 ADC 的软件转换启动功能
ADC_GetSoftwareStartConvStatus	获取 ADC 软件转换启动状态
ADC_DiscModeChannelCountConfig	对 ADC 规则组通道配置间断模式
ADC_DiscModeCmd	使能或者失能指定的 ADC 规则组通道的间断模式
ADC_RegularChannelConfig	设置指定 ADC 的规则组通道，设置它们的转化顺序和采样时间
ADC_ExternalTrigConvConfig	使能或者失能 ADCx 的经外部触发启动转换功能
ADC_GetConversionValue	返回最近一次 ADCx 规则组的转换结果
ADC_GetDuelModeConversionValue	返回最近一次双 ADC 模式下的转换结果
ADC_AutoInjectedConvCmd	使能或者失能指定 ADC 在规则组转化后自动开始注入组转换
ADC_InjectedDiscModeCmd	使能或者失能指定 ADC 的注入组间断模式
ADC_ExternalTrigInjectedConvConfig	配置 ADCx 的外部触发启动注入组转换功能
ADC_ExternalTrigInjectedConvCmd	使能或者失能 ADCx 的经外部触发启动注入组转换功能
ADC_SoftwareStartinjectedConvCmd	使能或者失能 ADCx 软件启动注入组转换功能
ADC_GetsoftwareStartinjectedConvStatus	获取指定 ADC 的软件启动注入组转换状态
ADC_InjectedChannleConfig	设置指定 ADC 的注入组通道，设置它们的转化顺序和采样时间
ADC_InjectedSequencerLengthConfig	设置注入组通道的转换序列长度
ADC_SetinjectedOffset	设置注入组通道的转换偏移值
ADC_GetInjectedConversionValue	返回 ADC 指定注入通道的转换结果
ADC_AnalogWatchdogCmd	使能或者失能指定单个/全体，规则/注入组通道上的模拟看门狗
ADC_AnalogWatchdongThresholdsConfig	设置模拟看门狗的高/低阈值
ADC_AnalogWatchdongSingleChannelConfig	对单个 ADC 通道设置模拟看门狗
ADC_TampSensorVrefintCmd	使能或者失能温度传感器和内部参考电压通道
ADC_GetFlagStatus	检查制定 ADC 标志位置 1 与否
ADC_ClearFlag	清除 ADCx 的待处理标志位
ADC_GetITStatus	检查指定的 ADC 中断是否发生
ADC_ClearITPendingBit	清除 ADCx 的中断待处理位

5. 备份寄存器（BKP）

备份寄存器由 10 个 16 位寄存器组成，可用来存储 20 个字节的用户应用程序数据。他们处在备份域里，当 VDD 电源被切断，他们仍然由 VBAT 维持供电。当系统在待机模式下被唤醒，或系统复位或电源复位时，他们也不会被复位。

此外，BKP 控制寄存器用来管理侵入检测和 RTC 校准功能。

Table 53. BKP 寄存器

寄存器	描述
DR 1-10	数据后备寄存器 1 到 10
RTCCR	RTC 时钟校准寄存器
CR	后备控制寄存器
CSR	后备控制状态寄存器

Table 54. BKP 库函数

函数名	描述
BKP_DeInit	将外设 BKP 的全部寄存器重设为缺省值
BKP_TamperPinLevelConfig	设置侵入检测管脚的有效电平
BKP_TamperPinCmd	使能或者失能管脚的侵入检测功能
BKP_ITConfig	使能或者失能侵入检测中断
BKP_RTCOutputConfig	选择在侵入检测管脚上输出的 RTC 时钟源
BKP_SetRTCCalibrationValue	设置 RTC 时钟校准值
BKP_WriteBackupRegister	向指定的后备寄存器中写入用户程序数据
BKP_ReadBackupRegister	从指定的后备寄存器中读出数据
BKP_GetFlagStatus	检查侵入检测管脚事件的标志位被设置与否
BKP_ClearFlag	清除侵入检测管脚事件的待处理标志位
BKP_GetITStatus	检查侵入检测中断发生与否
BKP_ClearITPendingBit	清除侵侵入检测中断的待处理位

6 控制器局域网（CAN）

本外设作为 CAN 网络的界面，支持 CAN 协议 2.0A 和 2.0B。它的设计目标是，以最小的 CPU 负荷来高效处理大量收到的报文。它也支持报文发送的优先级要求（优先级特性可软件配置）。

Section 6.1 描述了 CAN 固件函数库所使用的数据结构，Section 6.2 固件库函数介绍了函数库里的所有函数。

Table 70. CAN 寄存器

寄存器	描述
CAN_MCR	CAN 主控制寄存器
CAN_MSR	CAN 主状态寄存器
CAN_TSR	CAN 发送状态寄存器
CAN_RF0R	CAN 接收 FIFO 0 寄存器
CAN_RF1R	CAN 接收 FIFO 1 寄存器
CAN_IER	CAN 中断允许寄存器
CAN_ESR	CAN 错误状态寄存器
CAN_BTR	CAN 位时间特性寄存器
TIR	发送邮箱标识符寄存器
TDTR	发送邮箱数据长度和时间戳寄存器
TDLR	发送邮箱低字节数据寄存器
TDHR	发送邮箱高字节数据寄存器
RIR	接收 FIFO 邮箱标识符寄存器
RDTR	接收 FIFO 邮箱数据长度和时间戳寄存器
RDLR	接收 FIFO 邮箱低字节数据寄存器
RDHR	接收 FIFO 邮箱高字节数据寄存器
CAN_FMR	CAN 过滤器主控寄存器
CAN_FM0R	CAN 过滤器模式寄存器
CAN_FSC0R	CAN 过滤器位宽寄存器
CAN_FFA0R	CAN 过滤器 FIFO 关联寄存器
CAN_FA0R	CAN 过滤器激活寄存器
CAN_FR0	过滤器组 0 寄存器
CAN_FR1	过滤器组 1 寄存器

Table 71. CAN 库函数

函数名	描述
CAN_DeInit	将外设 CAN 的全部寄存器重设为缺省值
CAN_Init	根据 CAN_InitStruct 中指定的参数初始化外设 CAN 的寄存器
CAN_FilterInit	根据 CAN_FilterInitStruct 中指定的参数初始化外设 CAN 的寄存器
CAN_StructInit	把 CAN_InitStruct 中的每一个参数按缺省值填入
CAN_ITConfig	使能或者失能指定的 CAN 中断
CAN_Transmit	开始一个消息的传输
CAN_TransmitStatus	检查消息传输的状态
CAN_CancelTransmit	取消一个传输请求
CAN_FIFORelease	释放一个 FIFO
CAN_MessagePending	返回挂号的信息数量
CAN_Receive	接收一个消息
CAN_Sleep	使 CAN 进入低功耗模式
CAN_WakeUp	将 CAN 唤醒
CAN_GetFlagStatus	检查指定的 CAN 标志位被设置与否
CAN_ClearFlag	清除 CAN 的待处理标志位
CAN_GetITStatus	检查指定的 CAN 中断发生与否
CAN_ClearITPendingBit	清除 CAN 的中断待处理标志位

7 DMA控制器（DMA）

DMA 控制器提供 7 个数据通道的访问。由于外设实现了向存储器的映射，因此数据对来自或者发向外设的数据传输，也可以像内存之间的数据传输一样管理。

Table 104. DMA 寄存器

寄存器	描述
ISR	DMA 中断状态寄存器
IFCR	DMA 中断标志位清除寄存器
CCRx	DMA 通道 x 设置寄存器
CNDTRx	DMA 通道 x 待传输数据数目寄存器
CPARx	DMA 通道 x 外设地址寄存器
CMARx	DMA 通道 x 内存地址寄存器

Table 105. DMA 库函数

函数名	描述
DMA_DeInit	将 DMA 的通道 x 寄存器重设为缺省值
DMA_Init	根据 DMA_InitStruct 中指定的参数初始化 DMA 的通道 x 寄存器
DMA_StructInit	把 DMA_InitStruct 中的每一个参数按缺省值填入
DMA_Cmd	使能或者失能指定的通道 x
DMA_ITConfig	使能或者失能指定的通道 x 中断
DMA_GetCurrDataCounte	返回当前 DMA 通道 x 剩余的待传输数据数目
DMA_GetFlagStatus	检查指定的 DMA 通道 x 标志位设置与否
DMA_ClearFlag	清除 DMA 通道 x 待处理标志位
DMA_GetITStatus	检查指定的 DMA 通道 x 中断发生与否
DMA_ClearITPendingBit	清除 DMA 通道 x 中断待处理标志位

8 外部中断/事件控制器（EXTI）

外部中断/事件控制器由 19 个产生事件/中断要求的边沿检测器组成。每个输入线可以独立地配置输入类型（脉冲或挂起）和对应的触发事件（上升沿或下降沿或者双边沿都触发）。每个输入线都可以被独立的屏蔽。挂起寄存器保持着状态线的中断要求。

Table 128. EXTI 寄存器

寄存器	描述
IMR	中断屏蔽寄存器
EMR	事件屏蔽寄存器
RTSR	上升沿触发选择寄存器
FTSR	下降沿触发选择寄存器
SWIR	软件中断事件寄存器
PR	挂起寄存器

Table 129. EXTI 库函数

函数名	描述
EXTI_DeInit	将外设 EXTI 寄存器重设为缺省值
EXTI_Init	根据 EXTI_InitStruct 中指定的参数初始化外设 EXTI 寄存器
EXTI_StructInit	把 EXTI_InitStruct 中的每一个参数按缺省值填入
EXTI_GenerateSWInterrupt	产生一个软件中断
EXTI_GetFlagStatus	检查指定的 EXTI 线路标志位设置与否
EXTI_ClearFlag	清除 EXTI 线路挂起标志位
EXTI_GetITStatus	检查指定的 EXTI 线路触发请求发生与否
EXTI_ClearITPendingBit	清除 EXTI 线路挂起位

9 FLASH存储器(FLASH)

Table 142. FLASH 寄存器

寄存器	描述
ACR	FLASH 访问控制寄存器
KEYR	FPEC 密钥寄存器
OPTKEYR	选择字节密钥寄存器
SR	FLASH 状态寄存器
CR	FLASH 控制寄存器
AR	FLASH 地址寄存器
OBR	选择字节和状态寄存器
WRPR	选择字节写保护寄存器

Table 143. Option Byte（OB）寄存器

寄存器	描述
RDR	读出选择字节
USER	用户选择字节
Data0	Data0 选择字节
Data1	Data1 选择字节
WRP0	写保护 0 选择字节
WRP1	写保护 1 选择字节
WRP2	写保护 2 选择字节
WRP3	写保护 3 选择字节

Table 144. FLASH 库函数

函数名	描述
FLASH_SetLatency	设置代码延时值
FLASH_HalfCycleAccessCmd	使能或者失能 FLASH 半周期访问
FLASH_PrefetchBufferCmd	使能或者失能预取指缓存
FLASH_Unlock	解锁 FLASH 编写擦除控制器
FLASH_Lock	锁定 FLASH 编写擦除控制器
FLASH_ErasePage	擦除一个 FLASH 页面
FLASH_EraseAllPages	擦除全部 FLASH 页面
FLASH_EraseOptionBytes	擦除 FLASH 选择字节
FLASH_ProgramWord	在指定地址编写一个字
FLASH_ProgramHalfWord	在指定地址编写半字
FLASH_ProgramOptionByteData	在指定 FLASH 选择字节地址编写半字
FLASH_EnableWriteProtection	对期望的页面写保护
FLASH_ReadOutProtection	使能或者失能读出保护
FLASH_UserOptionByteConfig	编写 FLASH 用户选择字节：IWDG_SW /RST_STOP /RST_STDBY
FLASH_GetUserOptionByte	返回 FLASH 用户选择字节的值
FLASH_GetWriteProtectionOptionByte	返回 FLASH 写保护选择字节的值
FLASH_GetReadOutProtectionStatus	检查 FLASH 读出保护设置与否
FLASH_GetPrefetchBufferStatus	检查 FLASH 预取指缓存设置与否
FLASH_ITConfig	使能或者失能指定 FLASH 中断
FLASH_GetFlagStatus	检查指定的 FLASH 标志位设置与否
FLASH_ClearFlag	清除 FLASH 待处理标志位
FLASH_GetStatus	返回 FLASH 状态
FLASH_WaitForLastOperation	等待某一个 Flash 操作完成，或者发生 TIMEOUT

10 通用输入/输出（GPIO）

Table 178. GPIO 寄存器

寄存器	描述
CRL	端口配置低寄存器
CRH	端口配置高寄存器
IDR	端口输入数据寄存器
ODR	端口输出数据寄存器
BSRR	端口位设置/复位寄存器
BRR	端口位复位寄存器
LCKR	端口配置锁定寄存器
EVCR	事件控制寄存器
MAPR	复用重映射和调试 I/O 配置寄存器
EXTICR	外部中断线路 0-15 配置寄存器

Table 179. GPIO 库函数

函数名	描述
GPIO_DeInit	将外设 GPIOx 寄存器重设为缺省值
GPIO_AFIODeInit	将复用功能（重映射事件控制和 EXTI 设置）重设为缺省值
GPIO_Init	根据 GPIO_InitStruct 中指定的参数初始化外设 GPIOx 寄存器
GPIO_StructInit	把 GPIO_InitStruct 中的每一个参数按缺省值填入
GPIO_ReadInputDataBit	读取指定端口管脚的输入
GPIO_ReadInputData	读取指定的 GPIO 端口输入
GPIO_ReadOutputDataBit	读取指定端口管脚的输出
GPIO_ReadOutputData	读取指定的 GPIO 端口输出
GPIO_SetBits	设置指定的数据端口位
GPIO_ResetBits	清除指定的数据端口位
GPIO_WriteBit	设置或者清除指定的数据端口位
GPIO_Write	向指定 GPIO 数据端口写入数据
GPIO_PinLockConfig	锁定 GPIO 管脚设置寄存器
GPIO_EventOutputConfig	选择 GPIO 管脚用作事件输出
GPIO_EventOutputCmd	使能或者失能事件输出
GPIO_PinRemapConfig	改变指定管脚的映射
GPIO_EXTILineConfig	选择 GPIO 管脚用作外部中断线路

11 内部集成电路（I²C）

I2C 总线接口连接微控制器和串行 I2C 总线。它提供多主机功能，控制所有 I2C 总线特定的时序、协议、仲裁和定时。支持标准和快速两种模式，同时与 SMBus 2.0 兼容。I2C 总线有多种用途，包括 CRC 码的生成和校验、SMBus(系统管理总线 System Management Bus) PMBus(电源管理总线 Power Management Bus)。

I2C 驱动可以用来通过 I2C 界面发送和接收数据，还可以返回传输操作的状态。

Table 204. I2C 寄存器

寄存器	描述
CR1	I2C 控制寄存器 1
CR2	I2C 控制寄存器 2
OAR1	I2C 自身地址寄存器 1
OAR2	I2C 自身地址寄存器 2
DR	I2C 数据寄存器
SR1	I2C 状态寄存器 1
SR2	I2C 状态寄存器 2
CCR	I2C 时钟控制寄存器
TRISE	I2C 上升时间寄存器

Table 205. I2C 库函数

函数名	描述
I2C_DeInit	将外设 I2Cx 寄存器重设为缺省值
I2C_Init	根据 I2C_InitStruct 中指定的参数初始化外设 I2Cx 寄存器
I2C_StructInit	把 I2C_InitStruct 中的每一个参数按缺省值填入
I2C_Cmd	使能或者失能 I2C 外设
I2C_DMACmd	使能或者失能指定 I2C 的 DMA 请求
I2C_DMALastTransferCmd	使下一次 DMA 传输为最后一次传输
I2C_GenerateSTART	产生 I2Cx 传输 START 条件
I2C_GenerateSTOP	产生 I2Cx 传输 STOP 条件
I2C_AcknowledgeConfig	使能或者失能指定 I2C 的应答功能
I2C_OwnAddress2Config	设置指定 I2C 的自身地址 2
I2C_DualAddressCmd	使能或者失能指定 I2C 的双地址模式
I2C_GeneralCallCmd	使能或者失能指定 I2C 的广播呼叫功能
I2C_ITConfig	使能或者失能指定的 I2C 中断
I2C_SendData	通过外设 I2Cx 发送一个数据
I2C_ReceiveData	返回通过 I2Cx 最近接收的数据
I2C_Send7bitAddress	向指定的从 I2C 设备传送地址字
I2C_ReadRegister	读取指定的 I2C 寄存器并返回其值
I2C_SoftwareResetCmd	使能或者失能指定 I2C 的软件复位
I2C_SMBusAlertConfig	驱动指定 I2Cx 的 SMBusAlert 管脚电平为高或低
I2C_TransmitPEC	使能或者失能指定 I2C 的 PEC 传输
I2C_PECPositionConfig	选择指定 I2C 的 PEC 位置
I2C_CalculatePEC	使能或者失能指定 I2C 的传输字 PEC 值计算
I2C_GetPEC	返回指定 I2C 的 PEC 值
I2C_ARPCmd	使能或者失能指定 I2C 的 ARP
I2C_StretchClockCmd	使能或者失能指定 I2C 的时钟延展
I2C_FastModeDutyCycleConfig	选择指定 I2C 的快速模式占空比
I2C_GetLastEvent	返回最近一次 I2C 事件
I2C_CheckEvent	检查最近一次 I2C 事件是否是输入的事件
I2C_GetFlagStatus	检查指定的 I2C 标志位设置与否
I2C_ClearFlag	清除 I2Cx 的待处理标志位
I2C_GetITStatus	检查指定的 I2C 中断发生与否
I2C_ClearITPendingBit	清除 I2Cx 的中断待处理位

12 独立看门狗（IWDG）

独立看门狗（IWDG）用来解决应软件或者硬件引起的处理器故障。它也可以在停止（Stop）模式和待命（Standby）模式下工作。

Table 254. IWDG 寄存器

寄存器	描述
KR	IWDG 键值寄存器
PR	IWDG 预分频寄存器
RLR	IWDG 重装载寄存器
SR	IWDG 状态寄存器

Table 255. IWDG 库函数

函数名	描述
IWDG_WriteAccessCmd	使能或者失能对寄存器 IWDG_PR 和 IWDG_RLR 的写操作
IWDG_SetPrescaler	设置 IWDG 预分频值
IWDG_SetReload	设置 IWDG 重装载值
IWDG_ReloadCounter	按照 IWDG 重装载寄存器的值重装载 IWDG 计数器
IWDG_Enable	使能 IWDG
IWDG_GetFlagStatus	检查指定的 IWDG 标志位被设置与否

13 嵌套向量中断控制器（NVIC）

NVIC 驱动有多种用途：例如使能或者失能 IRQ 中断，使能或者失能单独的 IRQ 通道，改变 IRQ 通道的优先级等等。

Table 265. NVIC 寄存器

寄存器	描述
Enable	中断设置使能寄存器
Disable	中断清除使能寄存器
Set	中断设置待处理寄存器
Clear	中断清除待处理寄存器
Active	中断活动位寄存器
Priority	中断优先级寄存器
CPUID	CPU ID 基寄存器
IRQControlStatus	中断控制状态寄存器
ExceptionTableOffset	向量表移位寄存器
AIRC	应用控制/重置寄存器
SysCtrl	系统控制寄存器
ConfigCtrl	设置控制寄存器
SystemPriority	系统处理优先级寄存器
SysHandlerCtrl	系统处理控制和状态寄存器
ConfigFaultStatus	设置错误状态寄存器
HardFaultStatus	硬件错误状态寄存器
DebugFaultStatus	除错错误寄存器
MemorymanageFaultAddr	存储器管理错误地址寄存器
BusFaultAddr	总线错误地址寄存器

Table 266. NVIC 库函数

函数名	描述
NVIC_DeInit	将外设 NVIC 寄存器重设为缺省值
NVIC_SCBDeInit	将外设 SCB 寄存器重设为缺省值
NVIC_PriorityGroupConfig	设置优先级分组：先占优先级和从优先级
NVIC_Init	根据 NVIC_InitStruct 中指定的参数初始化外设 NVIC 寄存器
NVIC_StructInit	把 NVIC_InitStruct 中的每一个参数按缺省值填入
NVIC_SETPRIMASK	使能 PRIMASK 优先级：提升执行优先级至 0
NVIC_RESETPRIMASK	失能 PRIMASK 优先级
NVIC_SETFAULTMASK	使能 FAULTMASK 优先级：提升执行优先级至-1
NVIC_RESETFAULTMASK	失能 FAULTMASK 优先级
NVIC_BASEPRICONFIG	改变执行优先级从 N（最低可设置优先级）提升至 1
NVIC_GetBASEPRI	返回 BASEPRI 屏蔽值
NVIC_GetCurrentPendingIRQChannel	返回当前待处理 IRQ 标识符
NVIC_GetIRQChannelPendingBitStatus	检查指定的 IRQ 通道待处理位设置与否
NVIC_SetIRQChannelPendingBit	设置指定的 IRQ 通道待处理位
NVIC_ClearIRQChannelPendingBit	清除指定的 IRQ 通道待处理位
NVIC_GetCurrentActiveHandler	返回当前活动的 Handler（IRQ 通道和系统 Handler）的标识符
NVIC_GetIRQChannelActiveBitStatus	检查指定的 IRQ 通道活动位设置与否
NVIC_GetCPUID	返回 ID 号码，Cortex-M3 内核的版本号和实现细节
NVIC_SetVectorTable	设置向量表的位置和偏移
NVIC_GenerateSystemReset	产生一个系统复位
NVIC_GenerateCoreReset	产生一个内核（内核+NVIC）复位
NVIC_SystemLPConfig	选择系统进入低功耗模式的条件
NVIC_SystemHandlerConfig	使能或者失能指定的系统 Handler
NVIC_SystemHandlerPriorityConfig	设置指定的系统 Handler 优先级
NVIC_GetSystemHandlerPendingBitStatus	检查指定的系统 Handler 待处理位设置与否
NVIC_SetSystemHandlerPendingBit	设置系统 Handler 待处理位
NVIC_ClearSystemHandlerPendingBit	清除系统 Handler 待处理位
NVIC_GetSystemHandlerActiveBitStatus	检查系统 Handler 活动位设置与否
NVIC_GetFaultHandlerSources	返回表示出错的系统 Handler 源
NVIC_GetFaultAddress	返回产生表示出错的系统 Handler 所在位置的地址

14 功耗控制（PWR）

PWR 有多种用途，包括功耗管理和低功耗模式选择。

Table 321. PWR 寄存器

寄存器	描述
CR	功耗控制寄存器
CSR	功耗控制状态寄存器

Table 322. PWR 库函数

函数名	描述
PWR_DeInit	将外设 PWR 寄存器重设为缺省值
PWR_BackupAccessCmd	使能或者失能 RTC 和后备寄存器访问
PWR_PVDCmd	使能或者失能可编程电压探测器（PVD）
PWR_PVDLevelConfig	设置 PVD 的探测电压阈值
PWR_WakeUpPinCmd	使能或者失能唤醒管脚功能
PWR_EnterSTOPMode	进入停止（STOP）模式
PWR_EnterSTANDBYMode	进入待命（STANDBY）模式
PWR_GetFlagStatus	检查指定 PWR 标志位设置与否
PWR_ClearFlag	清除 PWR 的待处理标志位

15 复位和时钟设置（RCC）

Table 336. RCC 寄存器

寄存器	描述
CR	时钟控制寄存器
CFGR	时钟配置寄存器
CIR	时钟中断寄存器
APB2RSTR	APB2 外设复位寄存器
APB1RSTR	APB1 外设复位寄存器
AHBENR	AHB 外设时钟使能寄存器
APB2ENR	APB2 外设时钟使能寄存器
APB1ENR	APB1 外设时钟使能寄存器
BDCR	备份域控制寄存器
CSR	控制/状态寄存器

Table 337. RCC 库函数

函数名	描述
RCC_DeInit	将外设 RCC 寄存器重设为缺省值
RCC_HSEConfig	设置外部高速晶振（HSE）
RCC_WaitForHSEStartUp	等待 HSE 起振
RCC_AdjustHSICalibrationValue	调整内部高速晶振（HSI）校准值
RCC_HSICmd	使能或者失能内部高速晶振（HSI）
RCC_PLLConfig	设置 PLL 时钟源及倍频系数
RCC_PLLCmd	使能或者失能 PLL
RCC_SYSCLKConfig	设置系统时钟（SYSCLK）
RCC_GetSYSCLKSource	返回用作系统时钟的时钟源
RCC_HCLKConfig	设置 AHB 时钟（HCLK）
RCC_PCLK1Config	设置低速 AHB 时钟（PCLK1）
RCC_PCLK2Config	设置高速 AHB 时钟（PCLK2）
RCC_ITConfig	使能或者失能指定的 RCC 中断
RCC_USBCLKConfig	设置 USB 时钟（USBCLK）
RCC_ADCCLKConfig	设置 ADC 时钟（ADCCLK）
RCC_LSEConfig	设置外部低速晶振（LSE）
RCC_LSICmd	使能或者失能内部低速晶振（LSI）
RCC_RTCCLKConfig	设置 RTC 时钟（RTCCLK）
RCC_RTCCLKCmd	使能或者失能 RTC 时钟
RCC_GetClocksFreq	返回不同片上时钟的频率
RCC_AHBPeriphClockCmd	使能或者失能 AHB 外设时钟
RCC_APB2PeriphClockCmd	使能或者失能 APB2 外设时钟
RCC_APB1PeriphClockCmd	使能或者失能 APB1 外设时钟
RCC_APB2PeriphResetCmd	强制或者释放高速 APB（APB2）外设复位
RCC_APB1PeriphResetCmd	强制或者释放低速 APB（APB1）外设复位
RCC_BackupResetCmd	强制或者释放后备域复位
RCC_ClockSecuritySystemCmd	使能或者失能时钟安全系统
RCC_MCOConfig	选择在 MCO 管脚上输出的时钟源
RCC_GetFlagStatus	检查指定的 RCC 标志位设置与否
RCC_ClearFlag	清除 RCC 的复位标志位
RCC_GetITStatus	检查指定的 RCC 中断发生与否
RCC_ClearITPendingBit	清除 RCC 的中断待处理位

16 实时时钟（RTC）

RTC 提供了一系列连续工作的计数器，配合适当的软件，具有提供时钟-日历的功能。写入计数器的值可以设置整个系统的时间/日期。

Table 389. RTC 寄存器

寄存器	描述
CRH	控制寄存器高位
CRL	控制寄存器低位
PRLH	预分频装载寄存器高位
PRLL	预分频装载寄存器低位
DIVH	预分频分频因子寄存器高位
DIVL	预分频分频因子寄存器低位
CNTH	计数器寄存器高位
CNTL	计数器寄存器低位
ALRH	闹钟寄存器高位
ALRL	闹钟寄存器低位

Table 390. RTC 库函数

函数名	描述
RTC_ITConfig	使能或者失能指定的 RTC 中断
RTC_EnterConfigMode	进入 RTC 配置模式
RTC_ExitConfigMode	退出 RTC 配置模式
RTC_GetCounter	获取 RTC 计数器的值
RTC_SetCounter	设置 RTC 计数器的值
RTC_SetPrescaler	设置 RTC 预分频的值
RTC_SetAlarm	设置 RTC 闹钟的值
RTC_GetDivider	获取 RTC 预分频分频因子的值
RTC_WaitForLastTask	等待最近一次对 RTC 寄存器的写操作完成
RTC_WaitForSynchro	等待 RTC 寄存器(RTC_CNT, RTC_ALR and RTC_PRL)与 RTC 的 APB 时钟同步
RTC_GetFlagStatus	检查指定的 RTC 标志位设置与否
RTC_ClearFlag	清除 RTC 的待处理标志位
RTC_GetITStatus	检查指定的 RTC 中断发生与否
RTC_ClearITPendingBit	清除 RTC 的中断待处理位

17 串行外设接口（SPI）

串行外设接口（SPI）提供与外部设备进行同步串行通讯的功能。接口可以被设置工作在主模式或者从模式。

Table 407. SPI 寄存器

寄存器	描述
CR1	SPI 控制寄存器 1
CR2	SPI 控制寄存器 2
SR	SPI 状态寄存器
DR	SPI 数据寄存器
CRCPR	SPI CRC 多项式寄存器
RxCRCR	SPI 接收 CRC 寄存器
TxCRCR	SPI 发送 CRC 寄存器

Table 408. SPI 库函数

函数名	描述
SPI_DeInit	将外设 SPIx 寄存器重设为缺省值
SPI_Init	根据 SPI_InitStruct 中指定的参数初始化外设 SPIx 寄存器
SPI_StructInit	把 SPI_InitStruct 中的每一个参数按缺省值填入
SPI_Cmd	使能或者失能 SPI 外设
SPI_ITConfig	使能或者失能指定的 SPI 中断
SPI_DMACmd	使能或者失能指定 SPI 的 DMA 请求
SPI_SendData	通过外设 SPIx 发送一个数据
SPI_ReceiveData	返回通过 SPIx 最近接收的数据
SPI_DMALastTransferCmd	使下一次 DMA 传输为最后一次传输
SPI_NSSInternalSoftwareConfig	为选定的 SPI 软件配置内部 NSS 管脚
SPI_SSOutputCmd	使能或者失能指定的 SPI SS 输出
SPI_DataSizeConfig	设置选定的 SPI 数据大小
SPI_TransmitCRC	发送 SPIx 的 CRC 值
SPI_CalculateCRC	使能或者失能指定 SPI 的传输字 CRC 值计算
SPI_GetCRC	返回指定 SPI 的发送或者接受 CRC 寄存器值
SPI_GetCRCPolynomial	返回指定 SPI 的 CRC 多项式寄存器值
SPI_BiDirectionalLineConfig	选择指定 SPI 在双向模式下的数据传输方向
SPI_GetFlagStatus	检查指定的 SPI 标志位设置与否
SPI_ClearFlag	清除 SPIx 的待处理标志位
SPI_GetITStatus	检查指定的 SPI 中断发生与否
SPI_ClearITPendingBit	清除 SPIx 的中断待处理位

18 Cortex系统定时器（SysTick）

SysTick 提供 1 个 24 位、降序、零约束、写清除的计数器，具有灵活的控制机制。

Table 446. SysTick 寄存器

寄存器	描述
CTRL	SysTick 控制和状态寄存器
LOAD	SysTick 重装载值寄存器
VAL	SysTick 当前值寄存器
CALIB	SysTick 校准值寄存器

Table 447. SysTick 库函数

函数名	描述
SysTick_CLKSourceConfig	设置 SysTick 时钟源
SysTick_SetReload	设置 SysTick 重装载值
SysTick_CounterCmd	使能或者失能 SysTick 计数器
SysTick_ITConfig	使能或者失能 SysTick 中断
SysTick_GetCounter	获取 SysTick 计数器的值
SysTick_GetFlagStatus	检查指定的 SysTick 标志位设置与否

19 通用定时器（TIM）

通用定时器是一个通过可编程预分频器驱动的 16 位自动装载计数器构成。

它适用于多种场合，包括测量输入信号的脉冲长度(输入采集)或者产生输出波形(输出比较和 PWM)。

使用定时器预分频器和 RCC 时钟控制器预分频器，脉冲长度和波形周期可以在几个微秒到几个毫秒间调整。

Table 457. TIM 寄存器

寄存器	描述
CR1	控制寄存器 1
CR2	控制寄存器 2
SMCR	从模式控制寄存器
DIER	DMA/中断使能寄存器
SR	状态寄存器
EGR	事件产生寄存器
CCMR1	捕获/比较模式寄存器 1
CCMR2	捕获/比较模式寄存器 2
CCER	捕获/比较使能寄存器
CNT	计数器寄存器
PSC	预分频寄存器
APR	自动重装载寄存器
CCR1	捕获/比较寄存器 1
CCR2	捕获/比较寄存器 2
CCR3	捕获/比较寄存器 3
CCR4	捕获/比较寄存器 4
DCR	DMA 控制寄存器
DMAR	连续模式的 DMA 地址寄存器

Table 458. TIM 库函数

函数名	描述
TIM_DeInit	将外设 TIMx 寄存器重设为缺省值
TIM_TimeBaseInit	根据 TIM_TimeBaseInitStruct 中指定的参数初始化 TIMx 的时间基数单位
TIM_OCInit	根据 TIM_OCInitStruct 中指定的参数初始化外设 TIMx
TIM_ICInit	根据 TIM_ICInitStruct 中指定的参数初始化外设 TIMx
TIM_TimeBaseStructInit	把 TIM_TimeBaseInitStruct 中的每一个参数按缺省值填入
TIM_OCStructInit	把 TIM_OCInitStruct 中的每一个参数按缺省值填入
TIM_ICStructInit	把 TIM_ICInitStruct 中的每一个参数按缺省值填入
TIM_Cmd	使能或者失能 TIMx 外设
TIM _ITConfig	使能或者失能指定的 TIM 中断
TIM_DMAConfig	设置 TIMx 的 DMA 接口
TIM_DMACmd	使能或者失能指定的 TIMx 的 DMA 请求
TIM_InternalClockConfig	设置 TIMx 内部时钟
TIM_ITRxExternalClockConfig	设置 TIMx 内部触发为外部时钟模式
TIM_TIxExternalClockConfig	设置 TIMx 触发为外部时钟
TIM_ETRClockMode1Config	配置 TIMx 外部时钟模式 1
TIM_ETRClockMode2Config	配置 TIMx 外部时钟模式 2
TIM_ETRConfig	配置 TIMx 外部触发
TIM_SelectInputTrigger	选择 TIMx 输入触发源
TIM_PrescalerConfig	设置 TIMx 预分频
TIM_CounterModeConfig	设置 TIMx 计数器模式
TIM_ForcedOC1Config	置 TIMx 输出 1 为活动或者非活动电平
TIM_ForcedOC2Config	置 TIMx 输出 2 为活动或者非活动电平
TIM_ForcedOC3Config	置 TIMx 输出 3 为活动或者非活动电平
TIM_ForcedOC4Config	置 TIMx 输出 4 为活动或者非活动电平
TIM_ARRPreloadConfig	使能或者失能 TIMx 在 ARR 上的预装载寄存器
TIM_SelectCCDMA	选择 TIMx 外设的捕获比较 DMA 源
TIM_OC1PreloadConfig	使能或者失能 TIMx 在 CCR1 上的预装载寄存器
TIM_OC2PreloadConfig	使能或者失能 TIMx 在 CCR2 上的预装载寄存器
TIM_OC3PreloadConfig	使能或者失能 TIMx 在 CCR3 上的预装载寄存器
TIM_OC4PreloadConfig	使能或者失能 TIMx 在 CCR4 上的预装载寄存器
TIM_OC1FastConfig	设置 TIMx 捕获比较 1 快速特征

TIM_OC2FastConfig	设置 TIMx 捕获比较 2 快速特征
TIM_OC3FastConfig	设置 TIMx 捕获比较 3 快速特征
TIM_OC4FastConfig	设置 TIMx 捕获比较 4 快速特征
TIM_ClearOC1Ref	在一个外部事件时清除或者保持 OCREF1 信号
TIM_ClearOC2Ref	在一个外部事件时清除或者保持 OCREF2 信号
TIM_ClearOC3Ref	在一个外部事件时清除或者保持 OCREF3 信号
TIM_ClearOC4Ref	在一个外部事件时清除或者保持 OCREF4 信号
TIM_UpdateDisableConfig	使能或者失能 TIMx 更新事件
TIM_EncoderInterfaceConfig	设置 TIMx 编码界面
TIM_GenerateEvent	设置 TIMx 事件由软件产生
TIM_OC1PolarityConfig	设置 TIMx 通道 1 极性
TIM_OC2PolarityConfig	设置 TIMx 通道 2 极性
TIM_OC3PolarityConfig	设置 TIMx 通道 3 极性
TIM_OC4PolarityConfig	设置 TIMx 通道 4 极性
TIM_UpdateRequestConfig	设置 TIMx 更新请求源
TIM_SelectHallSensor	使能或者失能 TIMx 霍尔传感器接口
TIM_SelectOnePulseMode	设置 TIMx 单脉冲模式
TIM_SelectOutputTrigger	选择 TIMx 触发输出模式
TIM_SelectSlaveMode	选择 TIMx 从模式
TIM_SelectMasterSlaveMode	设置或者重置 TIMx 主/从模式
TIM_SetCounter	设置 TIMx 计数器寄存器值
TIM_SetAutoreload	设置 TIMx 自动重装载寄存器值
TIM_SetCompare1	设置 TIMx 捕获比较 1 寄存器值
TIM_SetCompare2	设置 TIMx 捕获比较 2 寄存器值
TIM_SetCompare3	设置 TIMx 捕获比较 3 寄存器值
TIM_SetCompare4	设置 TIMx 捕获比较 4 寄存器值
TIM_SetIC1Prescaler	设置 TIMx 输入捕获 1 预分频
TIM_SetIC2Prescaler	设置 TIMx 输入捕获 2 预分频
TIM_SetIC3Prescaler	设置 TIMx 输入捕获 3 预分频
TIM_SetIC4Prescaler	设置 TIMx 输入捕获 4 预分频
TIM_SetClockDivision	设置 TIMx 的时钟分割值
TIM_GetCapture1	获得 TIMx 输入捕获 1 的值
TIM_GetCapture2	获得 TIMx 输入捕获 2 的值
TIM_GetCapture3	获得 TIMx 输入捕获 3 的值
TIM_GetCapture4	获得 TIMx 输入捕获 4 的值
TIM_GetCounter	获得 TIMx 计数器的值
TIM_GetPrescaler	获得 TIMx 预分频值
TIM_GetFlagStatus	检查指定的 TIM 标志位设置与否
TIM_ClearFlag	清除 TIMx 的待处理标志位
TIM_GetITStatus	检查指定的 TIM 中断发生与否
TIM_ClearITPendingBit	清除 TIMx 的中断待处理位

20 高级控制定时器（TIM1）

高级控制定时器(TIM1)由一个 16 位的自动装载计数器组成，它由一个可编程预分频器驱动。

它适合多种用途，包含测量输入信号的脉冲宽度(输入捕获)，或者产生输出波形(输出比较，PWM，嵌入死区时间的互补 PWM 等)。

使用定时器预分频器和 RCC 时钟控制预分频器，可以实现脉冲宽度和波形周期从几个微秒到几个毫秒的调节。

21 通用同步异步收发器（USART）

通用同步异步收发器（USART）提供了一种灵活的方法来与使用工业标准 NRZ 异步串行数据格式的外部设备之间进行全双工数据交换。USART 利用分数波特率发生器提供宽范围的波特率选择。它支持同步单向通信和半双工单线通信。它也支持 LIN(局部互连网)，智能卡协议和 IrDA(红外数据组织)SIR ENDEC 规范，以及调制解调器(CTS/RTS)操作。它还允许多处理器通信。使用多缓冲器配置的 DMA 方式，可以实现高速数据通信。

Table 704. USART 寄存器

寄存器	描述
SR	USART 状态寄存器
DR	USART 数据寄存器
BRR	USART 波特率寄存器
CR1	USART 控制寄存器 1
CR2	USART 控制寄存器 2
CR3	USART 控制寄存器 3
GTPR	USART 保护时间和预分频寄存器

Table 705. USART 库函数

函数名	描述
USART_DeInit	将外设 USARTx 寄存器重设为缺省值
USART_Init	根据 USART_InitStruct 中指定的参数初始化外设 USARTx 寄存器
USART_StructInit	把 USART_InitStruct 中的每一个参数按缺省值填入
USART_Cmd	使能或者失能 USART 外设
USART_ITConfig	使能或者失能指定的 USART 中断
USART_DMACmd	使能或者失能指定 USART 的 DMA 请求
USART_SetAddress	设置 USART 节点的地址
USART_WakeUpConfig	选择 USART 的唤醒方式
USART_ReceiverWakeUpCmd	检查 USART 是否处于静默模式
USART_LINBreakDetectLengthConfig	设置 USART LIN 中断检测长度
USART_LINCmd	使能或者失能 USARTx 的 LIN 模式
USART_SendData	通过外设 USARTx 发送单个数据

USART_ReceiveData	返回 USARTx 最近接收到的数据
USART_SendBreak	发送中断字
USART_SetGuardTime	设置指定的 USART 保护时间
USART_SetPrescaler	设置 USART 时钟预分频
USART_SmartCardCmd	使能或者失能指定 USART 的智能卡模式
USART_SmartCardNackCmd	使能或者失能 NACK 传输
USART_HalfDuplexCmd	使能或者失能 USART 半双工模式
USART_IrDAConfig	设置 USART IrDA 模式
USART_IrDACmd	使能或者失能 USART IrDA 模式
USART_GetFlagStatus	检查指定的 USART 标志位设置与否
USART_ClearFlag	清除 USARTx 的待处理标志位
USART_GetITStatus	检查指定的 USART 中断发生与否
USART_ClearITPendingBit	清除 USARTx 的中断待处理位

22 窗口看门狗（WWDG）

窗口看门狗用来检测是否发生过软件错误。通常软件错误是由外部干涉或者不可预见的逻辑冲突引起的，这些错误将打断正常的程序流程。

Table 749. WWDG 寄存器

寄存器	描述
CR	WWDG 控制寄存器
CFR	WWDG 设置寄存器
SR	WWDG 状态寄存器

Table 750. WWDG 库函数

函数名	描述
WWDG_DeInit	将外设 WWDG 寄存器重设为缺省值
WWDG_SetPrescaler	设置 WWDG 预分频值
WWDG_SetWindowValue	设置 WWDG 窗口值
WWDG_EnableIT	使能 WWDG 早期唤醒中断（EWI）
WWDG_SetCounter	设置 WWDG 计数器值
WWDG_Enable	使能 WWDG 并装入计数器值
WWDG_GetFlagStatus	检查 WWDG 早期唤醒中断标志位被设置与否
WWDG_ClearFlag	清除早期唤醒中断标志位