



GW1NS-2C MCU

# 软件编程参考手册

RN516-1.2,2019-04-12

## **版权所有©2019 广东高云半导体科技股份有限公司**

未经本公司书面许可，任何单位和个人都不得擅自摘抄、复制、翻译本档内容的部分或全部，并不得以任何形式传播。

### **免责声明**

本档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除高云半导体在其产品的销售条款和条件中声明的责任之外，高云半导体概不承担任何法律或非法律责任。高云半导体对高云半导体产品的销售和 / 或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。高云半导体对档中包含的文字、图片及其它内容的准确性和完整性不承担任何法律或非法律责任，高云半导体保留修改档中任何内容的权利，恕不另行通知。高云半导体不承诺对这些档进行适时的更新。

## 版本信息

日期	版本	说明
2018/08/30	1.0	初始版本。
2018/11/30	1.1	<ul style="list-style-type: none"><li>● 优化软件编程库；</li><li>● 支持 uC/OS-III 和 FreeRTOS 操作系统。</li></ul>
2019/4/12	1.2	<ul style="list-style-type: none"><li>● 支持扩展外设 I2C、SPI 和 UART 软件编程；</li><li>● 更新 MCU 软件编程库；</li><li>● 更新 MCU 软件编程参考设计。</li></ul>

# 目录

目录 .....	i
图目录 .....	iv
表目录 .....	v
<b>1 软件编程库 .....</b>	<b>1</b>
1.1 MCU 软件编程库 .....	1
1.2 MCU 操作系统编程 .....	1
<b>2 存储系统 .....</b>	<b>3</b>
2.1 标准外设内存映射 .....	3
2.2 内核系统内存映射 .....	4
<b>3 中断处理 .....</b>	<b>5</b>
<b>4 通用异步收发器 .....</b>	<b>7</b>
4.1 特征 .....	7
4.2 寄存器定义 .....	8
4.3 初始化定义 .....	9
4.4 使用方法 .....	9
4.5 参考设计 .....	10
<b>5 定时器 .....</b>	<b>11</b>
5.1 特征 .....	11
5.2 寄存器定义 .....	11
5.3 初始化定义 .....	12
5.4 使用方法 .....	12
5.5 参考设计 .....	13
<b>6 看门狗 .....</b>	<b>14</b>

6.1 特征.....	14
6.2 寄存器定义.....	14
6.3 初始化定义.....	15
6.4 使用方法.....	15
6.5 参考设计.....	16
<b>7 通用输入输出.....</b>	<b>17</b>
7.1 特征.....	17
7.2 寄存器定义.....	18
7.3 初始化定义.....	20
7.4 使用方法.....	21
7.5 参考设计.....	21
<b>8 模拟/数字转换.....</b>	<b>23</b>
8.1 寄存器定义.....	23
8.2 初始化定义.....	23
8.3 使用方法.....	24
8.4 参考设计.....	24
<b>9 串行外设接口.....</b>	<b>25</b>
9.1 寄存器定义.....	25
9.2 初始化定义.....	25
9.3 使用方法.....	26
<b>10 中断监测器.....</b>	<b>27</b>
10.1 寄存器定义.....	27
10.2 使用方法.....	28
<b>11 系统控制器.....</b>	<b>30</b>
11.1 寄存器定义.....	30
11.2 使用方法.....	30
<b>12 I2C 总线接口.....</b>	<b>31</b>
12.1 寄存器定义.....	31
12.2 使用方法.....	31
12.3 参考设计.....	32
<b>13 软核扩展通用异步收发器.....</b>	<b>33</b>

---

13.1 特征 .....	33
13.2 寄存器定义.....	34
13.3 初始化定义.....	35
13.4 使用方法 .....	35
13.5 参考设计 .....	36
<b>14 SysTick .....</b>	<b>37</b>
14.1 特征.....	37
14.2 寄存器定义.....	37
14.3 使用方法 .....	37
14.4 参考设计 .....	38
<b>15 内存管理 .....</b>	<b>39</b>
15.1 特征 .....	39
15.2 使用方法 .....	39
15.3 参考设计 .....	39
<b>16 操作系统 .....</b>	<b>40</b>
16.1 uC/OS-III .....	40
16.1.1 特征 .....	40
16.1.2 操作系统编程 .....	40
16.1.3 操作系统配置 .....	41
16.1.4 参考设计 .....	41
16.2 FreeRTOS .....	41
16.2.1 特征.....	41
16.2.2 操作系统编程 .....	41
16.2.3 操作系统配置 .....	41
16.2.4 参考设计 .....	41

# 图目录

图 4-1 UART Buffering .....	8
图 5-1 TIMER .....	11
图 6-1 WatchDog Operation .....	14
图 7-1 GPIO Block.....	17
图 13-1 UART Buffering .....	34

# 表目录

表 1-1 GW1NS-2C MCU 软件编程库 .....	1
表 2-1 GW1NS-2C 标准外设内存映射 .....	3
表 2-2 系统控制内存映射 .....	4
表 3-1 GW1NS-2C 中断控制器 .....	5
表 4-1 UART 寄存器定义 .....	8
表 4-2 UART 初始化定义 .....	9
表 4-3 UART 使用方法 .....	9
表 5-1 TIMER 寄存器定义 .....	12
表 5-2 TIMER 初始化结构 .....	12
表 5-3 TIMER 使用方法 .....	12
表 6-1 WatchDog 寄存器定义 .....	14
表 6-2 WatchDog 初始化定义 .....	15
表 6-3 WatchDog 使用方法 .....	15
表 7-1 GPIO 寄存器定义 .....	18
表 7-2 GPIO 初始化定义 .....	20
表 7-3 GPIO 使用方法 .....	21
表 8-1 ADC 寄存器定义 .....	23
表 8-2 ADC 初始化定义 .....	23
表 8-3 ADC 使用方法 .....	24
表 9-1 SPI 寄存器定义 .....	25
表 9-2 SPI 初始化定义 .....	25
表 9-3 SPI 使用方法 .....	26
表 10-1 Interrupt Monitor 寄存器定义 .....	27
表 10-2 Interrupt Monitor 使用方法 .....	28
表 11-1 SYSCON 寄存器定义 .....	30
表 11-2 SYSCON 使用方法 .....	30



---

表 12-1 I2C 总线接口寄存器定义 .....	31
表 12-2 I2C 总线接口使用方法 .....	32
表 13-1 UART 寄存器定义 .....	34
表 13-2 UART 初始化定义 .....	35
表 13-3 UART 使用方法 .....	35
表 14-1 SysTick 寄存器定义 .....	37
表 14-2 SysTick 使用方法 .....	37

# 1 软件编程库

GW1NS-2C MCU 提供 MCU 软件编程库：

Gowin\_EMPU\_Src\c\_lib。

## 1.1 MCU 软件编程库

MCU 软件编程库，如表 1-1 所示。

表 1-1 GW1NS-2C MCU 软件编程库

文件	描述
startup_gw1ns2c.s	启动引导程序
core_cm3.c	ARM Cortex-M3 内核定义
gw1ns2c.h	寄存器定义和地址映射
system_gw1ns2c.c	系统初始化和系统时钟定义
gw1ns2c_flash.ld	Flash 链接脚本
gw1ns2c_adc.c	ADC 驱动函数定义
gw1ns2c_gpio.c	GPIO 驱动函数定义
gw1ns2c_uart.c	UART 驱动函数定义
gw1ns2c_timer.c	Timer 驱动函数定义
gw1ns2c_wdog.c	WatchDog 驱动函数定义
gw1ns2c_spi.c	SPI 驱动函数定义
gw1ns2c_i2c.c	I2C 驱动函数定义
gw1ns2c_misc.c	中断管理和 SysTick
gw1ns2c_syscon.c	系统控制驱动函数定义
gw1ns2c_it.c	中断定义

## 1.2 MCU 操作系统编程

MCU 支持 uC/OS-III 和 FreeRTOS 操作系统编程：

- uC/OS-III
- FreeRTOS

# 2 存储系统

## 2.1 标准外设内存映射

MCU 标准外设内存映射地址如表 2-1 所示。

表 2-1 GW1NS-2C 标准外设内存映射

标准外设	类型	地址映射	描述
Flash		0x00000000	128K 字节 Flash
Sram		0x20000000	2K、4K 或 8K 字节 BSRAM
TIMER0	TIMER_TypeDef	0x40000000	定时器 0
TIMER1	TIMER_TypeDef	0x40001000	定时器 1
UART0	UART_TypeDef	0x40004000	通用异步收发器 0
UART1	UART_TypeDef	0x40005000	通用异步收发器 1
WatchDog	WDOG_TypeDef	0x40008000	看门狗
GPIO0	GPIO_TypeDef	0x40010000	通用输入输出端口
SYSCON	SYSCON_TypeDef	0x4001F000	系统控制
I2C	I2C_TypeDef	0x40002000	I2C 总线接口
ADC	ADC_TypeDef	0x40002100	模拟/数字转换
SPI	SPI_TypeDef	0x40002200	串行外设接口
UART	UART_TypeDef	0x40002300	软核扩展通用异步收发器
Interrupt Monitor	IMONITOR_TypeDef	0x40002500	中断监测器

## 2.2 内核系统内存映射

MCU 内核系统内存映射如表 2-2 所示。

**表 2-2 系统控制内存映射**

系统控制	类型	地址映射	描述
ITM	ITM_Type	0xE0000000	ITM configuration struct
DWT	DWT_Type	0xE0001000	DWT configuration struct
CoreDebug	CoreDebug_Type	0xE000EDF0	Core Debug configuration struct
ETM	ETM_Type	0xE0041000	ETM configuration struct
SysTick	SysTick_Type	0xE000E010	SysTick configuration struct
NVIC	NVIC_BASE	0xE000E100	NVIC configuration struct
SCnSCB	SCnSCB_Type	0xE000E000	System control Register not in SCB
SCB	SCB_Type	0xE000ED00	SCB configuration struct
TPIU	TPIU_Type	0xE0040000	TPIU configuration struct

# 3 中断处理

嵌套向量中断控制器包括以下特征：

- 支持多达 24 个低延时中断
- 提供 2 条用户可用的中断处理信号（USER\_INT0 和 USER\_INT1）
- 支持 0-3 级可编程优先级
- 低延时中断和异常处理
- 中断信号边沿和脉冲检测
- 中断优先级动态调整

MCU 中断控制器如表 3-1 所示。

表 3-1 GW1NS-2C 中断控制器

Address	Interrupt	Number	Description
0x00000000	__StackTop		Top of Stack
0x00000004	Reset_Handler		Reset Handler
0x00000008	NMI_Handler		NMI Handler
0x0000000C	HardFault_Handler	-13	Hard Fault Handler
0x00000010	MemManage_Handler	-12	MPU Fault Handler
0x00000014	BusFault_Handler	-11	Bus Fault Handler
0x00000018	UsageFault_Handler	-10	Usage Fault Handler
0x0000001C	0		Reserved
0x00000020	0		Reserved
0x00000024	0		Reserved
0x00000028	0		Reserved
0x0000002C	SVC_Handler	-5	SVCcall Handler
0x00000030	DebugMon_Handler	-4	Debug Monitor Handler
0x00000034	0		Reserved
0x00000038	PendSV_Handler	-2	PendSV Handler
0x0000003C	SysTick_Handler	-1	SysTick Handler

Address	Interrupt	Number	Description
0x00000040	UART0_Handler	0	16+ 0: UART 0 RX and TX Handler
0x00000044	Spare1_Handler	1	16+ 1: Not Used
0x00000048	UART1_Handler	2	16+ 2: UART 1 RX and TX Handler
0x0000004C	Spare3_Handler	3	16+ 3: Not Used
0x00000050	Spare4_Handler	4	16+ 4: Not Used
0x00000054	0		16+ 5: Reserved
0x00000058	PORT0_COMB_Handler	6	16+ 6: GPIO Port 0 Combined Handler
0x0000005C	Spare7_Handler	7	16+ 7: Not Used
0x00000060	TIMER0_Handler	8	16+ 8: TIMER 0 handler
0x00000064	TIMER1_Handler	9	16+ 9: TIMER 1 handler
0x00000068	0		16+10: Reserved
0x0000006C	Spare11_Handler	11	16+11: Not Used
0x00000070	UARTOVF_Handler	12	16+12: UART 0,1 Overflow Handler
0x00000074	USER_INT0	13	16+13: USER_INT0 (default System Error Handler for FlashERR)
0x00000078	USER_INT1	14	16+14: USER_INT1 (default Embedded Flash Handler for FLASHINT)
0x0000007C	Spare15_Handler	15	16+ 15: Not Used
0x00000080	PORT0_0_Handler	16	16+16: GPIO Port 0 pin 0 Handler
0x00000084	PORT0_1_Handler	17	16+17: GPIO Port 0 pin 1 Handler
0x00000088	PORT0_2_Handler	18	16+18: GPIO Port 0 pin 2 Handler
0x0000008C	PORT0_3_Handler	19	16+19: GPIO Port 0 pin 3 Handler
0x00000090	PORT0_4_Handler	20	16+20: GPIO Port 0 pin 4 Handler
0x00000094	PORT0_5_Handler	21	16+21: GPIO Port 0 pin 5 Handler
0x00000098	PORT0_6_Handler	22	16+22: GPIO Port 0 pin 6 Handler
0x0000009C	PORT0_7_Handler	23	16+23: GPIO Port 0 pin 7 Handler
0x000000A0	PORT0_8_Handler	24	16+24: GPIO Port 0 pin 8 Handler
0x000000A4	PORT0_9_Handler	25	16+25: GPIO Port 0 pin 9 Handler
0x000000A8	PORT0_10_Handler	26	16+26: GPIO Port 0 pin 10 Handler
0x000000AC	PORT0_11_Handler	27	16+27: GPIO Port 0 pin 11 Handler
0x000000B0	PORT0_12_Handler	28	16+28: GPIO Port 0 pin 12 Handler
0x000000B4	PORT0_13_Handler	29	16+29: GPIO Port 0 pin 13 Handler
0x000000B8	PORT0_14_Handler	30	16+30: GPIO Port 0 pin 14 Handler
0x000000BC	PORT0_15_Handler	31	16+31: GPIO Port 0 pin 15 Handler

# 4 通用异步收发器

## 4.1 特征

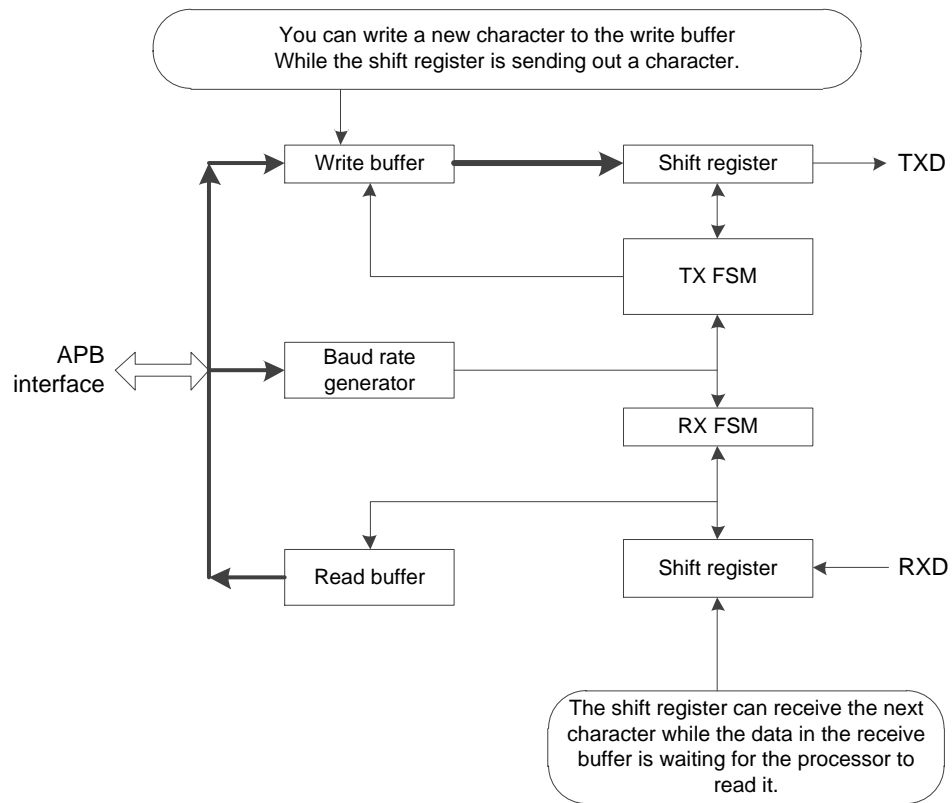
MCU 包含 2 个通过 APB1 总线访问的通用异步收发器 UART:

- 最大波特率为 921.6Kbit/s
- 无奇偶校验位
- 8 位数据位
- 1 位停止位

UART Buffering 如图 4-1 所示。



图 4-1 UART Buffering



UART 支持高速测试模式 HSTM (High Speed Test Mode)，当寄存器 CTRL[6] 设置为 1 时，串行数据每个周期传输 1 位，可以在很短时间内传输文本信息。

用户在使能 UART 时，必须设置波特率分频寄存器，例如，如果 APB1 总线频率运行在 12MHz，需要波特率为 9600，则可以设置波特率分频寄存器为  $12000000/9600=1250$ 。

## 4.2 寄存器定义

UART 寄存器定义，如表 4-1 所示。

表 4-1 UART 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
DATA	0x000	RW	8	0x--	[7:0] Data Value
STATE	0x004	RW	4	0x0	[3] RX buffer overrun, write 1 to clear [2] TX buffer overrun, write 1 to clear [1] RX buffer full, read-only [0] TX buffer full, read-only
CTRL	0x008	RW	7	0x00	[6] High speed test mode for TX only [5] RX overrun interrupt enable

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[4] TX overrun interrupt enable [3] RX interrupt enable [2] TX interrupt enable [1] RX enable [0] TX enable
INTSTATUS / INTCLEAR	0x00C	RW	4	0x0	[3] RX overrun interrupt,write 1 to clear [2] TX overrun interrupt,write 1 to clear [1] RX interrupt,write 1 to clear [0] TX interrupt,write 1 to clear
BAUDDIV	0x010	RW	20	0x00000	[19:0] Baud rate divider,the minimum number is 16

## 4.3 初始化定义

UART 初始化定义，如表 4-2 所示。

表 4-2 UART 初始化定义

名称	类型	数值	描述
UART_BaudRate	uint32_t	Max 921.6Kbit/s	Baud rate
UART_Mode	UARTMode_TypeDef	ENABLE/DISABLE	Enable/Disable TX/RX mode
UART_Int	UARTInt_TypeDef	ENABLE/DISABLE	Enable/Disable TX/RX interrupt
UART_Ovr	UARTOvr_TypeDef	ENABLE/DISABLE	Enable/Disable TX/RX overrun interrupt
UART_Hstm	FunctionalState	ENABLE/DISABLE	Enable/Disable TX high speed test mode

## 4.4 使用方法

UART 使用方法，如表 4-3 所示。

表 4-3 UART 使用方法

名称	描述
UART_Init	Initializes UARTx
UART_GetRxBufferFull	Returns UARTx RX buffer full status
UART_GetTxBufferFull	Returns UARTx TX buffer full status
UART_GetRxBufferOverrunStatus	Returns UARTx RX buffer overrun status
UART_GetTxBufferOverrunStatus	Returns UARTx TX buffer overrun status
UART_ClearRxBufferOverrunStatus	Clears Rx buffer overrun status
UART_ClearTxBufferOverrunStatus	Clears Tx buffer overrun status

名称	描述
UART_SendChar	Sends a character to UARTx TX buffer
UART_SendString	Sends a string to UARTx TX buffer
UART_ReceiveChar	Receives a character from UARTx RX buffer
UART_GetBaudDivider	Returns UARTx baud rate divider value
UART_GetTxIRQStatus	Returns UARTx TX interrupt status
UART_GetRxIRQStatus	Returns UARTx RX interrupt status
UART_ClearTxIRQ	Clears UARTx TX interrupt status
UART_ClearRxIRQ	Clears UARTx RX interrupt status
UART_GetTxOverrunIRQStatus	Returns UARTx TX overrun interrupt status
UART_GetRxOverrunIRQStatus	Returns UARTx RX overrun interrupt status
UART_ClearTxOverrunIRQ	Clears UARTx TX overrun interrupt request
UART_ClearRxOverrunIRQ	Clears UARTx RX overrun interrupt request
UART_SetHSTM	Sets UARTx TX high speed test mode
UART_ClrHSTM	Clears UARTx TX high speed test mode

## 4.5 参考设计

GW1NS-2C MCU 提供 ARM Keil MDK 和 GoWin GNU MCU Eclipse 软件环境的 UART 软件编程参考设计：

MCU\_RefDesign \Keil\_RefDesign\uart

MCU\_RefDesign\GNU\_RefDesign\uart

# 5 定时器

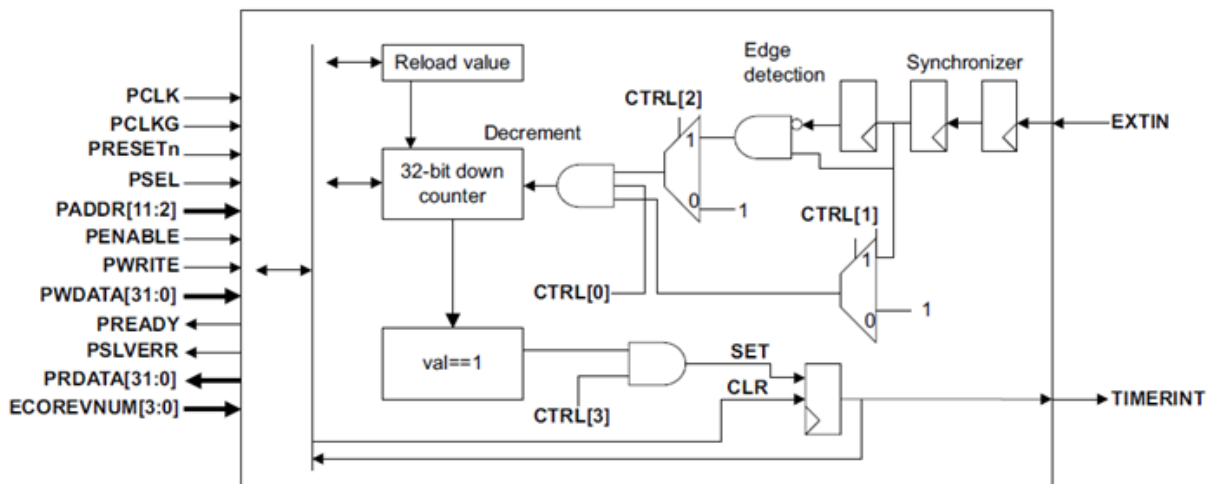
## 5.1 特征

MCU 包含 2 个通过 APB1 总线访问的同步标准定时器：

- 32 位计数器
- 可以产生中断请求信号
- 可以使用外部输入信号 EXTIN 使能时钟
- TIMER0: EXTIN 连接 GPIO[1]
- TIMER1: EXTIN 连接 GPIO[6]

TIMER 结构如图 5-1 所示。

图 5-1 TIMER



## 5.2 寄存器定义

TIMER 寄存器定义，如表 5-1 所示。

表 5-1 TIMER 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
CTRL	0x000	RW	4	0x0	[3] Timer interrupt enable [2] Select external input as clock [1] Select external input as enable [0] Enable
VALUE	0x004	RW	32	0x00000000	[31:0] Current value
RELOAD	0x008	RW	32	0x00000000	[31:0] Reload value, writing to this register sets the current value
INTSTATUS/ NTCLEAR	0x00C	RW	1	0x0	[0] Timer interrupt, write 1 to clear

## 5.3 初始化定义

TIMER 初始化定义，如表 5-2 所示。

表 5-2 TIMER 初始化结构

名称	类型	数值	描述
Reload	uint32_t		Reload value
TIMER_Int	TIMERInt_TypeDef	SET/RESET	Enable/Disable interrupt
TIMER_Exti	TIMERExti_TypeDef		External input as enable or clock

## 5.4 使用方法

TIMER 使用方法，如表 5-3 所示。

表 5-3 TIMER 使用方法

名称	描述
TIMER_Init	Initializes TIMERx
TIMER_StartTimer	Starts TIMERx
TIMER_StopTimer	Stops TIMERx
TIMER_GetIRQStatus	Returns TIMERx interrupt status
TIMER_ClearIRQ	Clears TIMERx interrupt status
TIMER_GetReload	Returns TIMERx reload value
TIMER_SetReload	Sets TIMERx reload value
TIMER_GetValue	Returns TIMERx current value
TIMER_SetValue	Sets TIMERx current value
TIMER_EnableIRQ	Enable TIMERx interrupt request
TIMER_DisableIRQ	Disable TIMERx interrupt request

## 5.5 参考设计

GW1NS-2C MCU 提供 ARM Keil MDK 和 GoWin GNU MCU Eclipse 软件环境的 Timer 软件编程参考设计:

MCU\_RefDesign\Keil\_RefDesign\timer

MCU\_RefDesign\GNU\_RefDesign\timer

# 6 看门狗

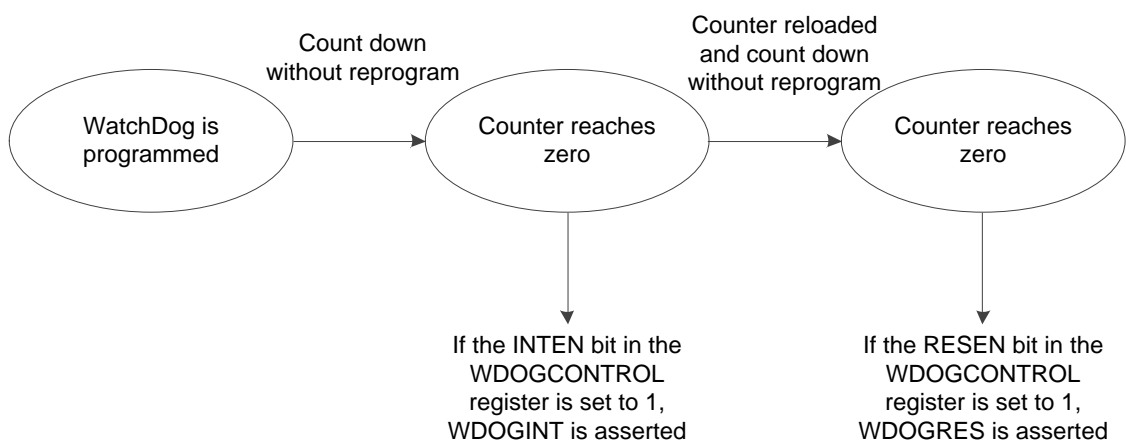
## 6.1 特征

MCU 包含 1 个通过 APB1 总线访问的看门狗 WatchDog:

- 基于由 LOAD 寄存器初始化的 32 位逐减计数器
- 产生中断请求
- 当时钟使能，由 WDOGCLK 信号上升沿触发计数器递减
- 监视中断，当计数器递减到 0 时，产生复位请求，计数器停止
- 响应软件崩溃引起的复位，提供软件恢复方法

WatchDog 操作如图 6-1 所示。

图 6-1 WatchDog Operation



## 6.2 寄存器定义

WatchDog 寄存器定义，如表 6-2 所示。

表 6-1 WatchDog 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
LOAD	0x00	RW	32	0xFFFFFFFF	The value from which the counter is to decrement
VALUE	0x04	RO	32	0xFFFFFFFF	The current value of the decrementing counter
CTRL	0x08	RW	2	0x0	[1] Enable reset output [0] Enable the interrupt
INTCLR	0x0C	WO			Clear the watchdog interrupt and reloads the counter
RIS	0x10	RO	1	0x0	Raw interrupt status from the counter
MIS	0x14	RO	1	0x0	Enable interrupt status from the counter
RESERVED	0xC00-0x014				Reserved
LOCK	0xC00	RW	32	0x00000000	[32:1] Enable register writes [0] Register write enable status
RESERVED	0xF00-0xC00				Reserved
ITCR	0xF00	RW	1	0x0	Integration test mode enable
ITOP	0xF04	WO	2	0x0	[1] Integration test WDOGRES value [0] Integration test WDOGINT value

## 6.3 初始化定义

WatchDog 初始化定义，如表 6-2 所示。

表 6-2 WatchDog 初始化定义

名称	类型	数值	描述
WDOG_Reload	uint32_t		Reload value
WDOG_Lock	WDOGLock_TypeDef	SET/RESET	Enable/Disable lock register write access
WDOG_Res	WDOGRes_TypeDef	SET/RESET	Enable/Disable reset flag
WDOG_Int	WDOGInt_TypeDef	SET/RESET	Enable/Disable interrupt flag
WDOG_ITMode	WDOGMode_Typedef	SET/RESET	Enable/Disable integration test mode flag

## 6.4 使用方法

WatchDog 使用方法，如表 6-3 所示。

表 6-3 WatchDog 使用方法

名称	描述
WDOG_Init	Initializes WatchDog



名称	描述
WDOG_RestartCounter	Restart watchdog counter
WDOG_GetCounterValue	Returns counter value
WDOG_SetResetEnable	Sets reset enable
WDOG_GetResStatus	Returns reset status
WDOG_SetIntEnable	Sets interrupt enable
WDOG_GetIntStatus	Returns interrupt enable
WDOG_ClrIntEnable	Clears interrupt enable
WDOG_GetRawIntStatus	Returns raw interrupt status
WDOG_GetMaskIntStatus	Returns masked interrupt status
WDOG_LockWriteAccess	Disable write access all registers
WDOG_UnlockWriteAccess	Enable write access all registers
WDOG_SetITModeEnable	Sets integration test mode enable
WDOG_ClrITModeEnable	Clears integration test mode enable
WDOG_GetITModeStatus	Returns integration test mode status
WDOG_SetITOP	Sets integration test output reset or interrupt
WDOG_GetITOPResStatus	Returns integration test output reset status
WDOG_GetITOPIntStatus	Returns integration test output interrupt status
WDOG_ClrITOP	Clears integration test output reset or interrupt

## 6.5 参考设计

GW1NS-2C MCU 提供 ARM Keil MDK 和 GoWin GNU MCU Eclipse 软件环境的 WatchDog 软件编程参考设计：

MCU\_RefDesign\Keil\_RefDesign\wdog

MCU\_RefDesign\GNU\_RefDesign\watchdog

# 7 通用输入输出

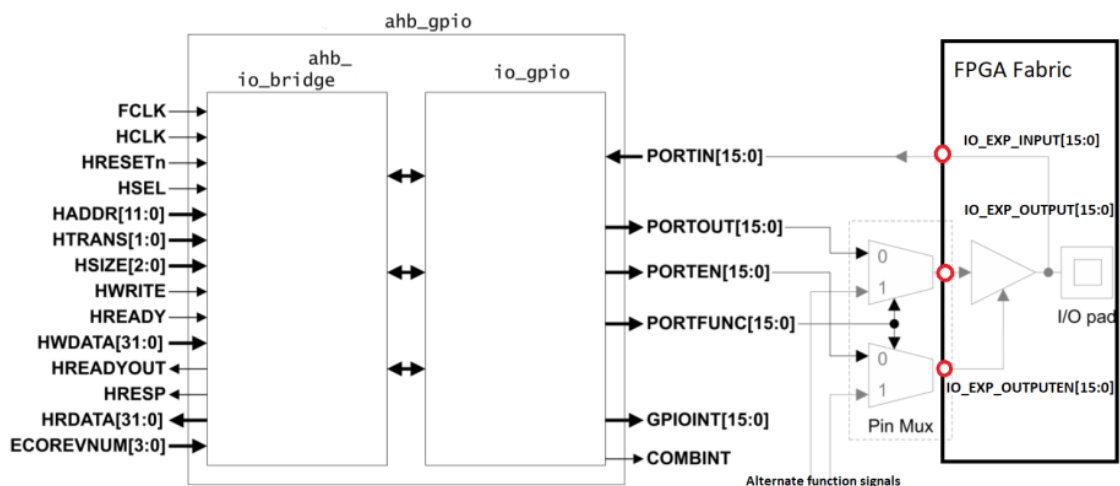
## 7.1 特征

MCU 包含 1 个通过 AHB 总线访问的 16 位输入输出接口的 GPIO 模块：

- 与 FPGA Fabric 连接；
- 每个 IO 管脚可以产生中断；
- 支持位掩码；
- 管脚复用功能。

GPIO 架构如图 7-1 所示。

图 7-1 GPIO Block



## 7.2 寄存器定义

GPIO 寄存器定义，如表 7-1 所示。

表 7-1 GPIO 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
DATA	0x0000	RW	16	0x----	[15:0] Data value Read Sampled at pin Write to data output register Read back value goes through double flip-flop synchronization logic with delay of two cycle
DATAOUT	0x0004	RW	16	0x0000	[15:0] Data output register value Read current value of data output register write to data output register
RESERVED	0x0008 -0x000C				Reserved
OUTENSET	0x0010	RW	16	0x0000	[15:0] Output enable set Write 1 to set the output enable bit Write 0 no effect Read back 0 indicates the signal direction as input 1 indicates the signal direction as output
OUTENCLR	0x0014	RW	16	0x0000	[15:0] Output enable clear Write 1 to clear the output enable bit Write 0 no effect Read back 0 indicates the signal direction as input 1 indicates the signal direction as output
ALTFUNCS ET	0x0018	RW	16	0x0000	[15:0] Alternative function set Write 1 to set the ALTFUNC bit Write 0 no effect Read back 0 for I/O 1 for an alternate function
ALTFUNCC LR	0x001C	RW	16	0x0000	[15:0] Alternative function clear Write 1 to clear the ALTFUNC bit Write 0 no effect Read back 0 for I/O 1 for an alternate function
INTENSET	0x0020	RW	16	0x0000	[15:0] Interrupt enable set Write 1 to set the enable bit

寄存器名称	地址偏移	类型	宽度	初始值	描述
					Write 0 no effect Read back 0 indicates interrupt disabled 1 indicates interrupt enabled
INTENCLR	0x0024	RW	16	0x0000	[15:0] Interrupt enable clear Write 1 to clear the enable bit Write 0 no effect Read back 0 indicates interrupt disabled 1 indicates interrupt enabled
INTTYPESET	0x0028	RW	16	0x0000	[15:0] Interrupt type set Write 1 to set the interrupt type bit Write 0 no effect Read back 0 for LOW/HIGH level 1 for falling edge or rising edge
INTTYPECLR	0x002C	RW	16	0x0000	[15:0] Interrupt type clear Write 1 to clear the interrupt type bit Write 0 no effect Read back 0 for LOW/HIGH level 1 for falling edge or rising edge
INTPOLSET	0x0030	RW	16	0x0000	[15:0] Polarity-level,edge IRQ config Write 1 to set the interrupt polarity bit Write 0 no effect Read back 0 for LOW level or falling edge 1 for HIGH level or rising edge
INTPOLCLR	0x0034	RW	16	0x0000	[15:0] Polarity-level,edge IRQ config Write 1 to clear the interrupt polarity bit Write 0 no effect Read back 0 for LOW level or falling edge 1 for HIGH level or rising edge
INTSTATUS /INTCLEAR	0x0038	RW	16	0x0000	[15:0] Write IRQ status clear register Write 1 to clear interrupt request Write 0 no effect Read back IRQ status register
MASKLOW BYTE	0x0400 -0x07FC	RW	16	0x----	Lower 8-bits masked access [9:2] of the address value are used as enable bit mask for the access [15:8] not used [7:0] Data for lower byte access,with [9:2] of address value used as enable mask for each bit
MASKHIGH BYTE	0x0800 -0x0BFC	RW	16	0x----	Higher 8-bits masked access [9:2] of the address value are used as enable

寄存器名称	地址偏移	类型	宽度	初始值	描述
					bit mask for the access [15:8] Data for higher byte access,with [9:2] of address value used as enable mask for each bit [7:0] not used
RESERVED	0x0C00 -0x0FCF				Reserved

## 7.3 初始化定义

GPIO 初始化定义，如表 7-2 所示。

表 7-2 GPIO 初始化定义

名称	类型	数值	描述
GPIO_Pin	uint32_t	GPIO_Pin_0 GPIO_Pin_1 GPIO_Pin_2 GPIO_Pin_3 GPIO_Pin_4 GPIO_Pin_5 GPIO_Pin_6 GPIO_Pin_7 GPIO_Pin_8 GPIO_Pin_9 GPIO_Pin_10 GPIO_Pin_11 GPIO_Pin_12 GPIO_Pin_13 GPIO_Pin_14 GPIO_Pin_15	16 bits GPIO Pins
GPIO_Mode	GPIO_Mode_TypeDef	GPIO_Mode_IN GPIO_Mode_OUT GPIO_Mode_AF	16 bits GPIO Pins mode
GPIO_Int	GPIO_Int_TypeDef	GPIO_Int_Disable GPIO_Int_Low_Level GPIO_Int_High_Level GPIO_Int_Falling_Edge GPIO_Int_Rising_Edge	16 bits GPIO Pins interrupt

## 7.4 使用方法

GPIO 使用方法，如表 7-3 所示。

表 7-3 GPIO 使用方法

名称	描述
GPIO_Init	Initializes GPIOx
GPIO_SetOutEnable	Sets GPIOx output enable
GPIO_ClrOutEnable	Clears GPIOx output enable
GPIO_GetOutEnable	Returns GPIOx output enable
GPIO_SetBit	GPIO output one
GPIO_ResetBit	GPIO output zero
GPIO_WriteBits	GPIO output
GPIO_ReadBits	GPIO input
GPIO_SetAltFunc	Sets GPIOx alternate function enable
GPIO_ClrAltFunc	Clears GPIOx alternate function enable
GPIO_GetAltFunc	Returns GPIOx alternate function enable
GPIO_IntClear	Clears GPIOx interrupt request
GPIO_GetIntStatus	Returns GPIOx interrupt status
GPIO_SetIntEnable	Sets GPIOx interrupt enable Returns GPIOx interrupt status
GPIO_ClrIntEnable	Clears GPIOx interrupt enable Returns GPIOx interrupt enable
GPIO_SetIntHighLevel	Setups GPIOx interrupt as high level
GPIO_SetIntRisingEdge	Setups GPIOx interrupt as rising edge
GPIO_SetIntLowLevel	Setups GPIOx interrupt as low level
GPIO_SetIntFallingEdge	Setups GPIOx interrupt as falling edge
GPIO_MaskedWrite	Setups GPIOx output value using masked access

## 7.5 参考设计

GW1NS-2C MCU 提供 ARM Keil MDK 和 GoWin GNU MCU Eclipse 软件环境的 GPIO 软件编程参考设计：

MCU\_RefDesign\Keil\_RefDesign\lcd

MCU\_RefDesign\Keil\_RefDesign\led

MCU\_RefDesign\Keil\_RefDesign\keyscan

MCU\_RefDesign\GNU\_RefDesign\lcd

MCU\_RefDesign\GNU\_RefDesign\led

MCU\_RefDesign\GNU\_RefDesign\keyscan

# 8 模拟/数字转换

## 8.1 寄存器定义

ADC 寄存器定义，如表 8-1 所示。

表 8-1 ADC 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
DATA	0x00	RO	12	0x000	[11:0] conversion data
STATUS	0x04	RW	2	0x0	[1] Start of conversion status [0] End of conversion status
CTRL	0x08	RW	6	0x00	[5] AD conversion mode [4] AD conversion starting [2:0] Channel

## 8.2 初始化定义

ADC 初始化定义，如表 8-2 所示。

表 8-2 ADC 初始化定义

名称	类型	数值	描述
ADC_Mode	uint32_t	ADC_MODE_CONT ADC_MODE_SINGLE	ADC continuous or single conversion mode
ADC_Status	uint32_t	ADC_STATUS_ON ADC_STATUS_OFF	Start or stop conversion
ADC_Chsel	uint32_t	ADC_CHSEL_0 ADC_CHSEL_1 ADC_CHSEL_2 ADC_CHSEL_3 ADC_CHSEL_4 ADC_CHSEL_5	ADC channel 0-7



名称	类型	数值	描述
		ADC_CHSEL_6 ADC_CHSEL_7	

## 8.3 使用方法

ADC 使用方法，如表 8-3 所示。

表 8-3 ADC 使用方法

名称	描述
ADC_Init	Initializes ADC
ADC_SetMode	Sets ADC conversion mode
ADC_GetMode	Returns ADC conversion mode
ADC_SetPowerStatus	Sets ADC running status
ADC_GetPowerStatus	Returns ADC running status
ADC_SetChannel	Sets ADC conversion channel
ADC_GetChannel	Returns ADC conversion channel
ADC_GetEocStatus	Returns stopping status
ADC_GetSocStatus	Returns starting status
ADC_ReadData	Returns ADC conversion data

## 8.4 参考设计

GW1NS-2C MCU 提供 ARM Keil MDK 和 GoWin GNU MCU Eclipse 软件环境的 ADC 软件编程参考设计：

MCU\_RefDesign\Keil\_RefDesign\adc

MCU\_RefDesign\Keil\_RefDesign\adc

# 9 串行外设接口

## 9.1 寄存器定义

SPI 寄存器定义，如表 9-1 所示。

表 9-1 SPI 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
RDATA	0x00	RO	8	0x00	Read data register
WDATA	0x04	WO	8	0x00	Write data register
STATUS	0x08	RW	8	0x00	[7] Error status [6] Receive ready status [5] Transmit ready status [4] Be transmitting [3] Transmit overrun error status [2] Receive overrun error status [1:0] Reserved
SSMASK	0x0C	RW	8	0x00	Unused selected slave address
CTRL	0x10	RW	5	0x00	[4:3] Clock selected [2] Polarity [1] Phase [0] Direction

## 9.2 初始化定义

SPI 初始化定义，如表 9-2 所示。

表 9-2 SPI 初始化定义

名称	类型	数值	描述
DIRECTION	FunctionalState	ENABLE/DISABLE	MSB/LSB first transmission
PHASE	FunctionalState	ENABLE/DISABLE	Posedge/Negedge transmit data

名称	类型	数值	描述
POLARITY	FunctionalState	ENABLE/DISABLE	Initialize ploarity to one/zero
CLKSEL	uint32_t	CLKSEL_CLK_DIV_2 CLKSEL_CLK_DIV_4 CLKSEL_CLK_DIV_6 CLKSEL_CLK_DIV_8	Select clock divided 2/4/6/8

## 9.3 使用方法

SPI 使用方法，如表 9-3 所示。

表 9-3 SPI 使用方法

名称	描述
SPI_Init	Initializes SPI
SPI_SetDirection	Sets direction
SPI_ClrDirection	Clears direction
SPI_GetDirection	Returns direction
SPI_SetPhase	Sets phase
SPI_ClrPhase	Clears phase
SPI_GetPhase	Returns phase
SPI_SetPolarity	Sets polarity
SPI_ClrPolarity	Clears polarity
SPI_GetPolarity	Returns polarity
SPI_SetClkSel	Sets clock selection
SPI_GetClkSel	Returns clock selection
SPI_GetToeStatus	Reads transmit overrun error status
SPI_GetRoeStatus	Reads receive overrun error status
SPI_GetTmtStatus	Reads transmitting status
SPI_GetTrdyStatu	Reads transmit ready status
SPI_GetRrdyStatus	Reads receive ready error status
SPI_GetErrStatus	Reads error status
SPI_ClrToeStatus	Clears transmit overrun error status
SPI_ClrRoeStatus	Clear receive overrun error status
SPI_ClrErrStatus	Clears error status
SPI_WriteData	Writes data
SPI_ReadData	Reads data

# 10 中断监测器

## 10.1 寄存器定义

Interrupt Monitor 寄存器定义，如表 10-1 所示。

表 10-1 Interrupt Monitor 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
INTSTATUS	0x00	RW	21	0x000000	[20] WatchDog interrupt status [19] UART1 interrupt status [18] UART0 interrupt status [17] TIMER1 interrupt status [16] TIMER0 interrupt status [15] GPIO0 Pin15 interrupt status [14] GPIO0 Pin14 interrupt status [13] GPIO0 Pin13 interrupt status [12] GPIO0 Pin12 interrupt status [11] GPIO0 Pin11 interrupt status [10] GPIO0 Pin10 interrupt status [9] GPIO0 Pin9 interrupt status [8] GPIO0 Pin8 interrupt status [7] GPIO0 Pin7 interrupt status [6] GPIO0 Pin6 interrupt status [5] GPIO0 Pin5 interrupt status [4] GPIO0 Pin4 interrupt status [3] GPIO0 Pin3 interrupt status [2] GPIO0 Pin2 interrupt status [1] GPIO0 Pin1 interrupt status [0] GPIO0 Pin0 interrupt status

## 10.2 使用方法

Interrupt Monitor 使用方法，如表 10-2 所示。

**表 10-2 Interrupt Monitor 使用方法**

名称	描述
IMONITOR_Init	Initializes IMONITOR
IMONITOR_SetGPIO0Pin0IntStatus	Sets GPIO0 Pin0 interrupt status
IMONITOR_GetGPIO0Pin0IntStatus	Returns GPIO0 Pin0 interrupt status
IMONITOR_ClrGPIO0Pin0IntStatus	Clears GPIO0 Pin0 interrupt status
IMONITOR_SetGPIO0Pin1IntStatus	Sets GPIO0 Pin1 interrupt status
IMONITOR_GetGPIO0Pin1IntStatus	Returns GPIO0 Pin1 interrupt status
IMONITOR_ClrGPIO0Pin1IntStatus	Clears GPIO0 Pin1 interrupt status
IMONITOR_SetGPIO0Pin2IntStatus	Sets GPIO0 Pin2 interrupt status
IMONITOR_GetGPIO0Pin2IntStatus	Returns GPIO0 Pin2 interrupt status
IMONITOR_ClrGPIO0Pin2IntStatus	Clears GPIO0 Pin2 interrupt status
IMONITOR_SetGPIO0Pin3IntStatus	Sets GPIO0 Pin3 interrupt status
IMONITOR_GetGPIO0Pin3IntStatus	Returns GPIO0 Pin3 interrupt status
IMONITOR_ClrGPIO0Pin3IntStatus	Clears GPIO0 Pin3 interrupt status
IMONITOR_SetGPIO0Pin4IntStatus	Sets GPIO0 Pin4 interrupt status
IMONITOR_GetGPIO0Pin4IntStatus	Returns GPIO0 Pin4 interrupt status
IMONITOR_ClrGPIO0Pin4IntStatus	Clears GPIO0 Pin4 interrupt status
IMONITOR_SetGPIO0Pin5IntStatus	Sets GPIO0 Pin5 interrupt status
IMONITOR_GetGPIO0Pin5IntStatus	Returns GPIO0 Pin5 interrupt status
IMONITOR_ClrGPIO0Pin5IntStatus	Clears GPIO0 Pin5 interrupt status
IMONITOR_SetGPIO0Pin6IntStatus	Sets GPIO0 Pin6 interrupt status
IMONITOR_GetGPIO0Pin6IntStatus	Returns GPIO0 Pin6 interrupt status
IMONITOR_ClrGPIO0Pin6IntStatus	Clears GPIO0 Pin6 interrupt status
IMONITOR_SetGPIO0Pin7IntStatus	Sets GPIO0 Pin7 interrupt status
IMONITOR_GetGPIO0Pin7IntStatus	Returns GPIO0 Pin7 interrupt status
IMONITOR_ClrGPIO0Pin7IntStatus	Clears GPIO0 Pin7 interrupt status
IMONITOR_SetGPIO0Pin8IntStatus	Sets GPIO0 Pin8 interrupt status
IMONITOR_GetGPIO0Pin8IntStatus	Returns GPIO0 Pin8 interrupt status
IMONITOR_ClrGPIO0Pin8IntStatus	Clears GPIO0 Pin8 interrupt status
IMONITOR_SetGPIO0Pin9IntStatus	Sets GPIO0 Pin9 interrupt status
IMONITOR_GetGPIO0Pin9IntStatus	Returns GPIO0 Pin9 interrupt status
IMONITOR_ClrGPIO0Pin9IntStatus	Clears GPIO0 Pin9 interrupt status
IMONITOR_SetGPIO0Pin10IntStatus	Sets GPIO0 Pin10 interrupt status
IMONITOR_GetGPIO0Pin10IntStatus	Returns GPIO0 Pin10 interrupt status
IMONITOR_ClrGPIO0Pin10IntStatus	Clears GPIO0 Pin10 interrupt status

名称	描述
IMONITOR_SetGPIO0Pin11IntStatus	Sets GPIO0 Pin11 interrupt status
IMONITOR_GetGPIO0Pin11IntStatus	Returns GPIO0 Pin11 interrupt status
IMONITOR_ClrGPIO0Pin11IntStatus	Clears GPIO0 Pin11 interrupt status
IMONITOR_SetGPIO0Pin12IntStatus	Sets GPIO0 Pin12 interrupt status
IMONITOR_GetGPIO0Pin12IntStatus	Returns GPIO0 Pin12 interrupt status
IMONITOR_ClrGPIO0Pin12IntStatus	Clears GPIO0 Pin12 interrupt status
IMONITOR_SetGPIO0Pin13IntStatus	Sets GPIO0 Pin13 interrupt status
IMONITOR_GetGPIO0Pin13IntStatus	Returns GPIO0 Pin13 interrupt status
IMONITOR_ClrGPIO0Pin13IntStatus	Clears GPIO0 Pin13 interrupt status
IMONITOR_SetGPIO0Pin14IntStatus	Sets GPIO0 Pin14 interrupt status
IMONITOR_GetGPIO0Pin14IntStatus	Returns GPIO0 Pin14 interrupt status
IMONITOR_ClrGPIO0Pin14IntStatus	Clears GPIO0 Pin14 interrupt status
IMONITOR_SetGPIO0Pin15IntStatus	Sets GPIO0 Pin15 interrupt status
IMONITOR_GetGPIO0Pin15IntStatus	Returns GPIO0 Pin15 interrupt status
IMONITOR_ClrGPIO0Pin15IntStatus	Clears GPIO0 Pin15 interrupt status
IMONITOR_SetTIMER0IntStatus	Sets TIMER0 interrupt status
IMONITOR_GetTIMER0IntStatus	Returns TIMER0 interrupt status
IMONITOR_ClrTIMER0IntStatus	Clears TIMER0 interrupt status
IMONITOR_SetTIMER1IntStatus	Sets TIMER0 interrupt status
IMONITOR_GetTIMER1IntStatus	Returns TIMER0 interrupt status
IMONITOR_ClrTIMER1IntStatus	Clears TIMER0 interrupt status
IMONITOR_SetUART0IntStatus	Sets UART0 interrupt status
IMONITOR_GetUART0IntStatus	Returns UART0 interrupt status
IMONITOR_ClrUART0IntStatus	Clears UART0 interrupt status
IMONITOR_SetUART1IntStatus	Sets UART1 interrupt status
IMONITOR_GetUART1IntStatus	Returns UART1 interrupt status
IMONITOR_ClrUART1IntStatus	Clears UART1 interrupt status
IMONITOR_SetWDOGIntStatus	Sets WatchDog interrupt status
IMONITOR_GetWDOGIntStatus	Returns WatchDog interrupt status
IMONITOR_ClrWDOGIntStatus	Clears WatchDog interrupt status

# 11 系统控制器

## 11.1 寄存器定义

SYSCON 寄存器定义，如表 11-1 所示。

表 11-1 SYSCON 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
REMAP	0x000	RW	1	0x0	Remap control register
PMUCTRL	0x004	RW	1	0x0	PMU control register
RESETOP	0x008	RW	1	0x0	reset option register
RESERVED0	0x00C				Reserved
RSTINFO	0x010	RW	3	0x0	[2] Lockup reset [1] Watchdog reset request [0] System reset request

## 11.2 使用方法

SYSCON 使用方法，如表 11-2 所示。

表 11-2 SYSCON 使用方法

名称	描述
SYSCON_Init	Initializes SYSCON
SYSCON_GetRemap	Returns REMAP
SYSCON_GetPmuctrlEnable	Returns PMUCTRL Enable
SYSCON_GetResetopLockuprst	Returns RESETOP LOCKUPRST
SYSCON_GetRstinfoSysresetreq	Returns RSTINFO SYSRESETREQ
SYSCON_GetRstinfoWdogresetreq	Returns RSTINFO SYSRESETREQ
SYSCON_GetRstinfoLockreset	Returns RSTINFO SYSRESETREQ

# 12 I2C 总线接口

## 12.1 寄存器定义

I2C 总线接口寄存器定义，如表 11-1 所示。

表 12-1 I2C 总线接口寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
PRER	0x00	RW	16	0xFFFF	Clock prescale register
CTR	0x04	RW	8	0x00	[7] Enable I2C [6] Enable I2C interrupt
TXR	0x08	WO	8	0x00	[7:1] Next transmission data [0] Data direction
RXR	0x0C	RO	8	0x00	[7:0] Last received data
CR	0x010	WO	8	0x00	[7] Start transmission status [6] Over transmission status [5] Read enable, read data from slave [4] Write enable, write data to slave [3] Acknowledge [0] Interrupt acknowledge
SR	0x14	RO	8	0x00	[7] Receive acknowledge signal from slave [6] I2C busy status [5] Arbitration loss [1] Data transmission status flag [0] Interrupt flag

## 12.2 使用方法

I2C 总线接口使用方法，如表 11-2 所示。



表 12-2 I2C 总线接口使用方法

名称	描述
I2C_Init	I2C Initialization
I2C_SendByte	Send a byte to I2C bus
I2C_SendBytes	Send multiple bytes to I2C bus
I2C_ReceiveByte	Read a byte from I2C bus
I2C_ReadBytes	Read multiple bytes from I2C bus
I2C_Rate_Set	Set I2C traffic rate
I2C_Enable	Enable I2C bus
I2C_UnEnable	Disable I2C bus

## 12.3 参考设计

GW1NS-2C MCU 提供 ARM Keil MDK 和 GoWin GNU MCU Eclipse 软件环境的 I2C 软件编程参考设计：

MCU\_RefDesign\Keil\_RefDesign\i2c

MCU\_RefDesign\GNU\_RefDesign\i2c

# 13 软核扩展通用异步收发器

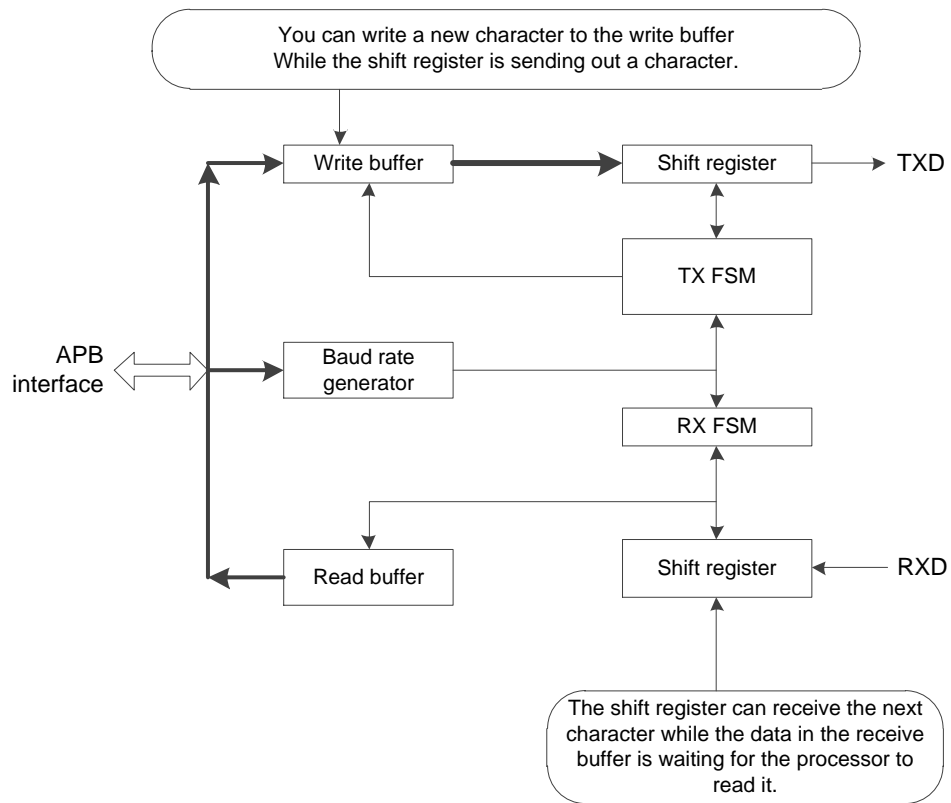
## 13.1 特征

MCU 包含 1 个通过 APB2 总线访问的软件扩展通用异步收发器 UART:

- 最大波特率为 921.6Kbit/s
- 无奇偶校验位
- 8 位数据位
- 1 位停止位

UART Buffering 如图 4-1 所示。

图 13-1 UART Buffering



UART 支持高速测试模式 HSTM (High Speed Test Mode)，当寄存器 CTRL[6] 设置为 1 时，串行数据每个周期传输 1 位，可以在很短时间内传输文本信息。

用户在使能 UART 时，必须设置波特率分频寄存器，例如，如果 APB1 总线频率运行在 12MHz，需要波特率为 9600，则可以设置波特率分频寄存器为  $12000000/9600=1250$ 。

## 13.2 寄存器定义

UART 寄存器定义，如表 4-1 所示。

表 13-1 UART 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
DATA	0x000	RW	8	0x--	[7:0] Data Value
STATE	0x004	RW	4	0x0	[3] RX buffer overrun, write 1 to clear [2] TX buffer overrun, write 1 to clear [1] RX buffer full, read-only [0] TX buffer full, read-only
CTRL	0x008	RW	7	0x00	[6] High speed test mode for TX only [5] RX overrun interrupt enable

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[4] TX overrun interrupt enable [3] RX interrupt enable [2] TX interrupt enable [1] RX enable [0] TX enable
INTSTATUS / INTCLEAR	0x00C	RW	4	0x0	[3] RX overrun interrupt,write 1 to clear [2] TX overrun interrupt,write 1 to clear [1] RX interrupt,write 1 to clear [0] TX interrupt,write 1 to clear
BAUDDIV	0x010	RW	20	0x00000	[19:0] Baud rate divider,the minimum number is 16

## 13.3 初始化定义

UART 初始化定义，如表 4-2 所示。

表 13-2 UART 初始化定义

名称	类型	数值	描述
UART_BaudRate	uint32_t	Max 921.6Kbit/s	Baud rate
UART_Mode	UARTMode_TypeDef	ENABLE/DISABLE	Enable/Disable TX/RX mode
UART_Int	UARTInt_TypeDef	ENABLE/DISABLE	Enable/Disable TX/RX interrupt
UART_Ovr	UARTOvr_TypeDef	ENABLE/DISABLE	Enable/Disable TX/RX overrun interrupt
UART_Hstm	FunctionalState	ENABLE/DISABLE	Enable/Disable TX high speed test mode

## 13.4 使用方法

UART 使用方法，如表 4-3 所示。

表 13-3 UART 使用方法

名称	描述
UART_Init	Initializes UARTx
UART_GetRxBufferFull	Returns UARTx RX buffer full status
UART_GetTxBufferFull	Returns UARTx TX buffer full status
UART_GetRxBufferOverrunStatus	Returns UARTx RX buffer overrun status
UART_GetTxBufferOverrunStatus	Returns UARTx TX buffer overrun status
UART_ClearRxBufferOverrunStatus	Clears Rx buffer overrun status
UART_ClearTxBufferOverrunStatus	Clears Tx buffer overrun status

名称	描述
UART_SendChar	Sends a character to UARTx TX buffer
UART_SendString	Sends a string to UARTx TX buffer
UART_ReceiveChar	Receives a character from UARTx RX buffer
UART_GetBaudDivider	Returns UARTx baud rate divider value
UART_GetTxIRQStatus	Returns UARTx TX interrupt status
UART_GetRxIRQStatus	Returns UARTx RX interrupt status
UART_ClearTxIRQ	Clears UARTx TX interrupt status
UART_ClearRxIRQ	Clears UARTx RX interrupt status
UART_GetTxOverrunIRQStatus	Returns UARTx TX overrun interrupt status
UART_GetRxOverrunIRQStatus	Returns UARTx RX overrun interrupt status
UART_ClearTxOverrunIRQ	Clears UARTx TX overrun interrupt request
UART_ClearRxOverrunIRQ	Clears UARTx RX overrun interrupt request
UART_SetHSTM	Sets UARTx TX high speed test mode
UART_ClrHSTM	Clears UARTx TX high speed test mode

## 13.5 参考设计

GW1NS-2C MCU 提供 ARM Keil MDK 和 GoWin GNU MCU Eclipse 软件环境的软核扩展 UART 软件编程参考设计：

MCU\_RefDesign \Keil\_RefDesign\uartes

MCU\_RefDesign\GNU\_RefDesign\uartes

# 14SysTick

## 14.1 特征

MCU 提供一个 24 位系统节拍定时器 SysTick，具有自动重载和溢出中断功能，MCU 可以通过这个定时器获得一定的时间间隔。

## 14.2 寄存器定义

SysTick 寄存器定义，如表 11-1 所示。

表 14-1 SysTick 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
CTRL	0x00	RW	17	0x000000	[16] Count down to zero flag [2] External clock source [1] Enable interrupt request [0] Enable SysTick
LOAD	0x04	RW	24	0x0000000	[23:0] Reload value
VAL	0x08	RW	24	0x0000000	[23:0] Return current count down value
CALIB	0x0C	RW	32	0x00000000	[31] Whether a separate reference clock is provided [30] Whether the TENMS value is exact [23:0] 10ms calibration value

## 14.3 使用方法

SysTick 使用方法，如表 11-2 所示。

表 14-2 SysTick 使用方法

名称	描述
SysTick_Config	Initialize and start the SysTick counter and interrupt

## 14.4 参考设计

GW1NS-2C MCU 提供 ARM Keil MDK 和 GoWin GNU MCU Eclipse 软件环境的 SysTick 软件编程参考设计:

MCU\_RefDesign\Keil\_RefDesign\systick

MCU\_RefDesign\GNU\_RefDesign\systick

# 15 内存管理

## 15.1 特征

GW1NS-2C MCU 提供动态内存管理方法，可以动态申请与释放内存。

## 15.2 使用方法

内存管理使用方法，如表 11-2 所示。

表 15-1 内存管理使用方法

名称	描述
mem_init	Initialize memory management
mymemset	Set the values of memory space
mymemcpy	Copy from source to destination
mymemcmp	Compare source with destination
mymalloc	Allocate a memory space dynamically
myfree	Free a memory space

## 15.3 参考设计

GW1NS-2C MCU 提供 ARM Keil MDK 和 GoWin GNU MCU Eclipse 软件环境的内存管理软件编程参考设计：

MCU\_RefDesign\Keil\_RefDesign\mm

MCU\_RefDesign\GNU\_RefDesign\mm



# 16 操作系统

MCU 支持 uC/OS-III 和 FreeRTOS 操作系统编程。

## 16.1 uC/OS-III

### 16.1.1 特征

- uC/OS-III 是一个可扩展的，可固化的，抢占式的实时内核，管理的任务个数不受限制
- uC/OS-III 是第三代内核，提供了现代实时内核所期望的功能，包括资源管理、同步、任务间通信等
- uC/OS-III 提供了很多其它实时内核所没有的特性，比如能在运行时测量运行性能，直接发送信号或消息给任务，任务能同时等待多个信号量和消息队列
- 高云提供已成功移植高云开发环境的 uC/OS-III 参考设计，或者 uC/OS-III 官网 <http://www.micrium.com> 下载

### 16.1.2 操作系统编程

MCU 软件编程参考设计使用的 uC/OS-III 版本为 V3.03.00。

uC/OS-III 操作系统包括：

- uC-CPU
- uC-LIB
- UCOS\_BSP
- uCOS\_CONFIG
- uCOS-III

### 16.1.3 操作系统配置

用户可以通过修改 UCOSIII\_CONFIG\os\_cfg.h 和 os\_cfg\_app.h 来配置 uC/OS-III。

### 16.1.4 参考设计

GW1NS-2C MCU 提供 ARM Keil MDK 和 GoWin GNU MCU Eclipse 软件环境的 uC/OS-III 软件编程参考设计：

MCU\_RefDesign\Keil\_RefDesign\ucos\_iii

MCU\_RefDesign\GNU\_RefDesign\ucos\_iii

## 16.2 FreeRTOS

### 16.2.1 特征

- FreeRTOS 是一个迷你的实时操作系统
- FreeRTOS 作为一个轻量级的操作系统，功能包括：任务管理、时间管理、信号量、消息队列、内存管理、记录功能、软件定时器、协程等，可基本满足较小系统的需要
- FreeRTOS 操作系统是完全免费的操作系统，具有源码公开、可移植、可裁减、调度策略灵活的特点
- 高云提供已成功移植高云开发环境的 FreeRTOS 参考设计，或者 FreeRTOS 官网 <http://www.FreeRTOS.org> 下载

### 16.2.2 操作系统编程

MCU 软件编程参考设计使用的 FreeRTOS 版本为 V9.0.0。

### 16.2.3 操作系统配置

用户可以修通过改 include\FreeRTOSConfig.h 来配置 FreeRTOS。

### 16.2.4 参考设计

GW1NS-2C MCU 提供 ARM Keil MDK 和 GoWin GNU MCU Eclipse 软件环境的 FreeRTOS 软件编程参考设计：

MCU\_RefDesign\Keil\_RefDesign\free\_rtos

MCU\_RefDesign\GNU\_RefDesign\free\_rtos

