Gowin_EMPU_M1 软件编程

参考手册

IPUG533-2.1,2024-07-12

免责声明

本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止反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。高云半导体对文档中包含的文字、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文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文档进行适时的更新。

版本信息

日期	版本	说明
2019/02/19	1.0	初始版本。
2019/07/18	1.1	MCU 软件编程设计支持扩展外部设备 CAN、Ethernet、SPI-Flash、RTC、DualTimer、TRNG、I2C、SPI、SD-Card。
2019/08/18	1.2	● MCU 硬件设计与软件编程设计支持扩展外部设备 DDR3 Memory； ● 修复已知 ITCM、DTCM Size 和 IDE 问题。
2019/09/27	1.3	● MCU 硬件设计与软件编程设计支持外部设备 SPI-Flash 的读、写和擦除功能； ● MCU 软件编程设计支持外部设备 I2C 一次连续多字节读、写功能； ● 修复已知 MCU 软件编程设计中 AHB2 扩展接口和 APB2 扩展接口地址映射问题； ● 修复已知 MCU 软件编程设计中 DDR3 Memory 连续读、写问题。
2020/01/16	1.4	● MCU 硬件设计与软件编程设计支持外部设备 PSRAM； ● 更新 MCU 编译软件 GMD V1.0； ● 更新 RTOS 参考设计； ● 增加 AHB2 和 APB2 扩展总线接口硬件和软件参考设计。
2020/03/05	1.5	MCU 软件编程设计支持外部设备 SD-Card 的读、写功能。
2020/06/12	1.6	● MCU 支持外部指令存储器； ● MCU 支持外部数据存储器； ● 扩展 6 个 AHB 总线接口； ● 扩展 16 个 APB 总线接口； ● GPIO 支持多种接口类型； ● I^2C 支持多种接口类型。
2021/07/16	1.7	● 修复已知的 SPI 全双工读写问题； ● 更新 MCU 软件版本。
2023/05/11	1.8	● 支持扩展的外部中断输入； ● 支持 Arora V FPGA 产品； ● 更新软件编程开发库； ● 更新 Bootloader； ● 更新软件编程参考设计； ● 新增 RT-Thread Nano 软件参考设计； ● 新增 TCP/IP 协议栈软件参考设计。
2023/07/21	1.9	更新已测试软件版本。
2024/03/07	2.0	● 更新软件编程库； ● 更新软件编程参考设计。
2024/07/12	2.1	● 更新 SPI Flash Memory 驱动函数； ● 支持 QSPI Flash Memory 驱动函数。

目录

图目录

表目录

1 软件编程库

Gowin_EMPU_M1 提供软件编程库：…\library。

通过此链接获取软件编程库：

cdn.gowinsemi.com.cn/Gowin_EMPU_M1.zip

Gowin_EMPU_M1 提供以下几种软件编程方法：

- 单片机软件编程
- 嵌入式 RTOS 软件编程
- 协议栈软件编程

1.1 单片机软件编程

Gowin_EMPU_M1 软件编程库，提供单片机软件编程方法，如表 1-1 所示。

表 1-1 单片机软件编程

函数库文件	描述
系统定义	
startup_GOWIN_M1.s	Cortex-M1内核启动引导程序
core_cm1.h	Cortex-M1内核寄存器定义
GOWIN_M1.h	中断向量表、外设寄存器和地址映射定义
system_GOWIN_M1.c system_GOWIN_M1.h	Cortex-M1内核系统初始化和系统时钟定义
Linker定义	
GOWIN_M1_flash_burn.ld GOWIN_M1_flash_xip.ld	GMD Flash链接器： burn：Flash引导启动，ITCM运行； xip：Flash引导启动和运行。
外设驱动定义	
GOWIN_M1_gpio.c GOWIN_M1_gpio.h	GPIO驱动函数定义
GOWIN_M1_can.c GOWIN_M1_can.h	CAN驱动函数定义

函数库文件	描述
GOWIN_M1_ethernet.c GOWIN_M1_ethernet.h	Ethernet驱动函数定义
GOWIN_M1_ddr3.c GOWIN_M1_ddr3.h	DDR3 Memory驱动函数定义
GOWIN_M1_psram.c GOWIN_M1_psram.h	PSRAM Memory驱动函数定义
GOWIN_M1_spi_flash.c GOWIN_M1_spi_flash.h	SPI-Flash Memory读、写和擦除功能驱动函数定义
GOWIN_M1_qspi_flash.c GOWIN_M1_qspi_flash.h	QSPI-Flash Memory驱动函数定义
GOWIN_M1_timer.c GOWIN_M1_timer.h	Timer驱动函数定义
GOWIN_M1_wdog.c GOWIN_M1_wdog.h	Watch Dog驱动函数定义
GOWIN_M1_uart.c GOWIN_M1_uart.h	UART驱动函数定义
GOWIN_M1_rtc.c GOWIN_M1_rtc.h	RTC驱动函数定义
GOWIN_M1_trng.c GOWIN_M1_trng.h	TRNG驱动函数定义
GOWIN_M1_dualtimer.c GOWIN_M1_dualtimer.h	DualTimer驱动函数定义
GOWIN_M1_i2c.c GOWIN_M1_i2c.h	I2C Master驱动函数定义
GOWIN_M1_spi.c GOWIN_M1_spi.h	SPI Master驱动函数定义
GOWIN_M1_sdcard.c GOWIN_M1_sdcard.h	SD-Card驱动函数定义
GOWIN_M1_misc.c GOWIN_M1_misc.h	中断优先级管理和SysTick定义
GOWIN_M1_it.c GOWIN_M1_it.h	中断处理函数定义
中间件定义	
retarget.c uart.c uart.h	UART printf函数重定义； UART初始化定义
malloc.c malloc.h	动态内存管理
gpio.c gpio.h	GPIO初始化定义
delay.c delay.h	延时函数定义

1.2 嵌入式 RTOS 软件编程

Gowin_EMPU_M1 支持以下几种嵌入式 RTOS 软件编程：

- uC/OS-III
- FreeRTOS
- RT-Thread Nano 版本

1.3 协议栈软件编程

Gowin_EMPU_M1 支持以太网 TCP/IP 协议栈软件编程，提供开源 LwIP 协议栈和 uIP 协议栈。

LwIP 协议栈是一个小型开源的 TCP/IP 协议栈，有无操作系统都可以运行，能够在保持 TCP 协议主要功能的基础上减少对内存的占用。

uIP 协议栈是一个简单的嵌入式网络协议栈，其不需要操作系统的支持，以事件驱动的方式编程，实现了四个基本协议 ARP、IP、ICMP 和 TCP。

2 存储系统

2.1 标准外设内存映射

Gowin_EMPU_M1 标准外设内存映射地址定义，如表 2-1 所示。

表 2-1 标准外设内存映射定义

标准外设	类型	地址映射	描述
ITCM	-	0x00000000	1KB, 2KB, 4KB, 8KB, 16KB, 32KB, 64KB, 128KB, 256KB, 512KB
DTCM	-	0x20000000	1KB, 2KB, 4KB, 8KB, 16KB, 32KB, 64KB, 128KB, 256KB, 512KB
External Instruction Memory	-	0x00000000	外部指令存储器
External Data Memory	-	0x20100000	外部数据存储器
TIMER0	TIMER_TypeDef	0x50000000	定时器0（Timer0）
TIMER1	TIMER_TypeDef	0x50001000	定时器1（Timer1）
UART0	UART_TypeDef	0x50004000	通用异步收发器0（UART0）
UART1	UART_TypeDef	0x50005000	通用异步收发器1（UART1）
Watch Dog	WDOG_TypeDef	0x50008000	看门狗（Watch Dog）
RTC	RTC_RegDef	0x50006000	实时时钟（RTC）
TRNG	TRNG_RegDef	0x5000F000	真随机数发生器（TRNG）
DualTimer	DUALTIMER_RegDef	0x50002000	双重定时器（DualTimer）
SPI_FLASH	SPI_FLASH_RegDef	0x50003000	串行外设接口闪存（SPI-Flash和QSPI-Flash Memory）
I2C	I2C_TypeDef	0x5000A000	内部集成电路总线（I²C Master）
SPI	SPI_TypeDef	0x5000B000	串行外设接口（SPI Master）
SD-Card	SDCard_TypeDef	0x50009000	安全数字存储卡（SD-Card）
GPIO0	GPIO_TypeDef	0x40000000	通用输入输出端口（GPIO）
CAN	CAN_RegDef	0x45000000	控制器局域网络（CAN）
Ethernet	ETH_RegDef	0x46000000	以太网（Ethernet）

标准外设	类型	地址映射	描述
DDR3	DDR3_RegDef	0x88000000	三代双倍数据率同步动态随机存储器（DDR3 Memory）
PSRAM	PSRAM_TypeDef	0x82000000	伪静态随机访问存储器（PSRAM Memory）
APB Master [1]	-	0x60000000	扩展APB Master [1]
APB Master [2]	-	0x60100000	扩展APB Master [2]
APB Master [3]	-	0x60200000	扩展APB Master [3]
APB Master [4]	-	0x60300000	扩展APB Master [4]
APB Master [5]	-	0x60400000	扩展APB Master [5]
APB Master [6]	-	0x60500000	扩展APB Master [6]
APB Master [7]	-	0x60600000	扩展APB Master [7]
APB Master [8]	-	0x60700000	扩展APB Master [8]
APB Master [9]	-	0x60800000	扩展APB Master [9]
APB Master [10]	-	0x60900000	扩展APB Master [10]
APB Master [11]	-	0x60A00000	扩展APB Master [11]
APB Master [12]	-	0x60B00000	扩展APB Master [12]
APB Master [13]	-	0x60C00000	扩展APB Master [13]
APB Master [14]	-	0x60D00000	扩展APB Master [14]
APB Master [15]	-	0x60E00000	扩展APB Master [15]
APB Master [16]	-	0x60F00000	扩展APB Master [16]
AHB Master [1]	-	0x80000000	扩展AHB Master [1]
AHB Master [2]	-	0x81000000	扩展AHB Master [2]
AHB Master [3]	-	0x86000000	扩展AHB Master [3]
AHB Master [4]	-	0x89000000	扩展AHB Master [4]
AHB Master [5]	-	0x8A000000	扩展AHB Master [5]
AHB Master [6]	-	0x8B000000	扩展AHB Master [6]

2.2 内核系统内存映射

Cortex-M1 内核系统内存映射定义，如表 2-2 所示。

表 2-2 内核系统内存映射定义

系统控制	类型	地址映射	描述
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

3 中断处理

Gowin_EMPU_M1 嵌套向量中断控制器，包括以下特征：

- 提供最多 32 个用户可用的中断处理信号，用户可配置 1 或 8 或 16 或 32 个外部中断处理信号；
- 提供 4 个扩展的外部中断信号；
- 支持 0-3 级可编程优先级。

Cortex-M1 中断向量表定义，如表 3-1 所示。

表 3-1 中断向量表定义

中断向量地址	中断类型	中断号	描述
0x00000000	__StackTop	–	Top of Stack
0x00000004	Reset_Handler	–	Reset Handler
0x00000008	NMI_Handler	-14	NMI Handler
0x0000000C	HardFault_Handler	-13	Hard Fault Handler
0x00000010	0	–	Reserved
0x00000014	0	–	Reserved
0x00000018	0	–	Reserved
0x0000001C	0	–	Reserved
0x00000020	0	–	Reserved
0x00000024	0	–	Reserved
0x00000028	0	–	Reserved
0x0000002C	SVC_Handler	-5	SVCall Handler
0x00000030	0	–	Reserved
0x00000034	0	–	Reserved
0x00000038	PendSV_Handler	-2	PendSV Handler
0x0000003C	SysTick_Handler	-1	SysTick Handler
0x00000040	UART0_Handler	0	16+ 0: UART 0 RX and TX Handler
0x00000044	UART1_Handler	1	16+ 1: UART 1 RX and TX Handler
0x00000048	TIMER0_Handler	2	16+ 2: Timer 0 Handler

中断向量地址	中断类型	中断号	描述
0x0000004C	TIMER1_Handler	3	16+ 3: Timer 1 Handler
0x00000050	GPIO0_Handler	4	16+ 4: GPIO Port 0 Combined Handler
0x00000054	UARTOVF_Handler	5	16+ 5: UART 0,1 Overflow Handler
0x00000058	RTC_Handler	6	16+ 6: RTC Handler
0x0000005C	I2C_Handler	7	16+ 7: I2C Handler
0x00000060	CAN_Handler	8	16+ 8: CAN Handler
0x00000064	ETH_Handler	9	16+ 9: ETH Handler
0x00000068	EXTINT_0_Handler	10	16+10: External 0 Handler
0x0000006C	DTimer_Handler	11	16+11: DualTimer Handler
0x00000070	TRNG_Handler	12	16+12: TRNG Handler
0x00000074	EXTINT_1_Handler	13	16+13: External 1 Handler
0x00000078	EXTINT_2_Handler	14	16+14: External 2 Handler
0x0000007C	EXTINT_3_Handler	15	16+15: External 3 Handler
0x00000080	GPIO0_0_Handler	16	16+16: GPIO0_0 Handler
0x00000084	GPIO0_1_Handler	17	16+17: GPIO0_1 Handler
0x00000088	GPIO0_2_Handler	18	16+18: GPIO0_2 Handler
0x0000008C	GPIO0_3_Handler	19	16+19: GPIO0_3 Handler
0x00000090	GPIO0_4_Handler	20	16+20: GPIO0_4 Handler
0x00000094	GPIO0_5_Handler	21	16+21: GPIO0_5 Handler
0x00000098	GPIO0_6_Handler	22	16+22: GPIO0_6 Handler
0x0000009C	GPIO0_7_Handler	23	16+23: GPIO0_7 Handler
0x000000A0	GPIO0_8_Handler	24	16+24: GPIO0_8 Handler
0x000000A4	GPIO0_9_Handler	25	16+25: GPIO0_9 Handler
0x000000A8	GPIO0_10_Handler	26	16+26: GPIO0_10 Handler
0x000000AC	GPIO0_11_Handler	27	16+27: GPIO0_11 Handler
0x000000B0	GPIO0_12_Handler	28	16+28: GPIO0_12 Handler
0x000000B4	GPIO0_13_Handler	29	16+29: GPIO0_13 Handler
0x000000B8	GPIO0_14_Handler	30	16+30: GPIO0_14 Handler
0x000000BC	GPIO0_15_Handler	31	16+31: GPIO0_15 Handler

4 UART

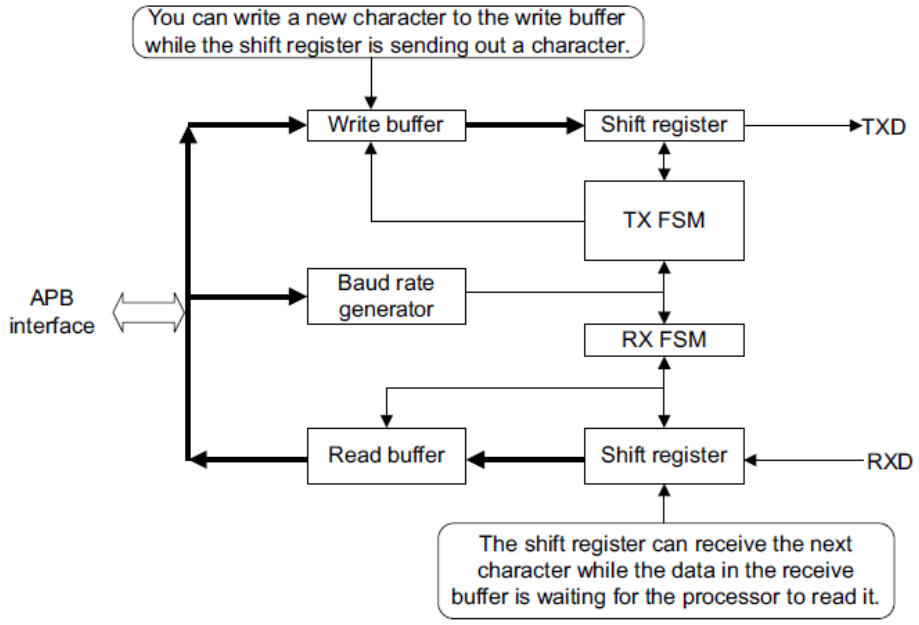
4.1 特征

Gowin_EMPU_M1，包含 2 个通过 APB 总线访问的 UART 外设：

- 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 结构框图，如图 4-1 所示。

图 4-1 UART 结构框图



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。UART 寄存器定义位于 library\libraries\cmsis\cm1\device_support\GOWIN_M1.h。

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 [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。UART 初始化定义位于 library\libraries\drivers\inc\GOWIN_M1_uart.h。

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

名称	类型	数值	描述
UART_Ovr	UARTOvr_TypeDef	ENABLE/DISABLE	Enable/Disable TX/RX overrun interrupt
UART_Hstm	FunctionalState	ENABLE/DISABLE	Enable/Disable TX hisgh speed test mode

4.4 驱动程序使用方法

UART 驱动程序使用方法，如表 4-3 所示。UART 驱动程序定义位于 library\libraries\drivers\src\GOWIN_M1_uart.c。

表 4-3 UART 驱动程序使用方法

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
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

5 Timer

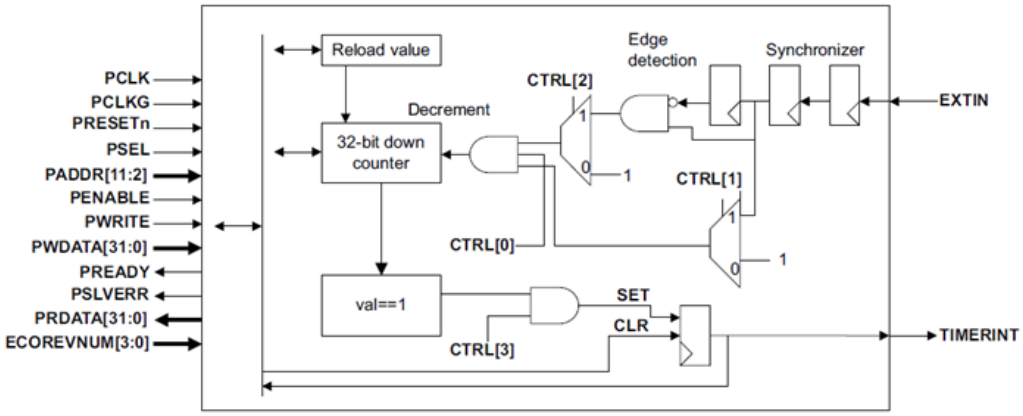
5.1 特征

Gowin_EMPU_M1，包含 2 个通过 APB 总线访问的同步标准定时器外设：

- 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

Timer 结构框图，如图 5-1 所示。

图 5-1 Timer 结构框图



5.2 寄存器定义

Timer 寄存器定义，如表 5-1 所示。Timer 寄存器定义位于 library\libraries\cmsis\cm1\device_support\GOWIN_M1.h。

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

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[0] Enable
VALUE	0x004	RW	32	0x0000 0000	[31:0] Current value
RELOAD	0x008	RW	32	0x0000 0000	[31:0] Reload value, writing to this register sets the current value
INTSTATUS/INTCLEAR	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。Timer 初始化定义位于 library\libraries\drivers\inc\GOWIN_M1_timer.h。

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 驱动程序使用方法

Timer 驱动程序使用方法，如表 5-3 所示。Timer 驱动程序定义位于 library\libraries\drivers\src\GOWIN_M1_timer.c。

表 5-3 Timer 驱动程序使用方法

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

6 Watch Dog

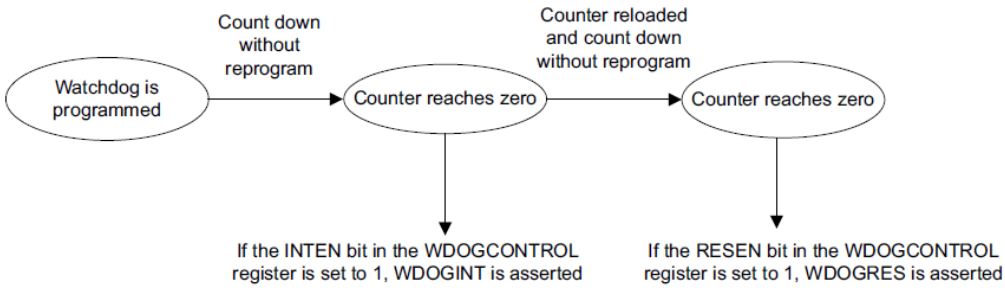
6.1 特征

Gowin_EMPU_M1，包含 1 个通过 APB 总线访问的 Watch Dog 外设：

- 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。

Watch Dog 操作流程，如图 6-1 所示。

图 6-1 Watch Dog 操作流程



6.2 寄存器定义

Watch Dog 寄存器定义，如表 6-1 所示。Watch Dog 寄存器定义位于 library\libraries\cmsis\cm1\device_support\GOWIN_M1.h。

表 6-1 Watch Dog 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	0xFFFF FFFF	The value from which the counter is to decrement
VALUE	0x04	RO	32	0xFFFF FFFF	The current value of the decrementing counter

寄存器名称	地址偏移	类型	宽度	初始值	描述
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	0x0000 0000	[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

Watch Dog 初始化定义，如表 6-2 所示。Watch Dog 初始化定义位于 library\libraries\drivers\GOWIN_M1_wdog.h。

表 6-2 Watch Dog 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 驱动程序使用方法

Watch Dog 驱动程序使用方法，如表 6-3 所示。Watch Dog 驱动程序定义位于 library\libraries\drivers\src\GOWIN_M1_wdog.c。

表 6-3 Watch Dog 驱动程序使用方法

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

7 GPIO

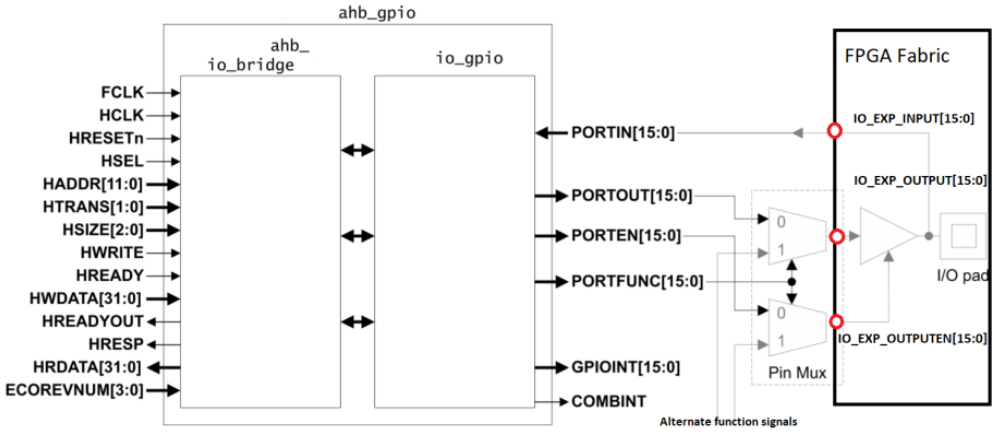
7.1 特征

Gowin_EMPU_M1，包含 1 个通过 AHB 总线访问的 16 位输入输出接口的 GPIO 外设：

- 与 FPGA Fabric 连接
- 支持位掩码
- 管脚复用功能

GPIO 结构框图，如图 7-1 所示。

图 7-1 GPIO 结构框图



7.2 寄存器定义

GPIO 寄存器定义，如表 7-1 所示。GPIO 寄存器定义位于 library\libraries\cmsis\cm1\device_support\GOWIN_M1.h。

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

寄存器名称	地址偏移	类型	宽度	初始值	描述
					Synchronize 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 intput. 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 intput. 1 indicates the signal direction as output.
ALTFUNCSET	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
ALTFUNCCLR	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 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

寄存器名称	地址偏移	类型	宽度	初始值	描述
					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 18ffect Read back IRQ status register
MASKLOWBYTE	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

寄存器名称	地址偏移	类型	宽度	初始值	描述
					value used as enable mask for each bit
MASKHIGHBYTE	0x0800 -0x0BFC	RW	16	0x----	Higher 8-bits masked access [9:2] of the address value are used as enable 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。GPIO 初始化定义位于 library\libraries\drivers\inc\GOWIN_M1_gpio.h。

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	GPIOMode_TypeDef	GPIO_Mode_IN GPIO_Mode_OUT GPIO_Mode_AF	16 bits GPIO Pins mode
GPIO_Int	GPIOInt_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 驱动程序使用方法

GPIO 驱动程序使用方法，如表 7-3 所示。GPIO 驱动程序定义位于 library\libraries\drivers\src\GOWIN_M1_gpio.c。

表 7-3 GPIO 驱动程序使用方法

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

8 I²C Master

8.1 特征

Gowin_EMPU_M1，包含一个通过 APB 总线访问的 I²C Master 外设：

- APB 总线接口
- 符合业界标准的 I²C 总线协议
- 总线仲裁及仲裁丢失检测
- 总线忙状态检测
- 产生中断标志
- 产生起始、终止、重复起始和应答信息
- 支持起始、终止和重复起始检测
- 支持 7 位寻址模式

8.2 寄存器定义

I²C Master 寄存器定义，如表 8-1 所示。I²C Master 寄存器定义位于 library\libraries\cmsis\cm1\device_support\GOWIN_M1.h。

表 8-1 I²C Master 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	32	0x0000FFFF	Clock prescale register [31:15] Reserved [15:0] Prescale value = sys_clk/(5*SCL)-1
CTR	0x04	RW	32	0x00000000	[31:8] Reserved [7] Enable I2C function [6] Enable I2C interrupt [5:0] Reserved
TXR	0x08	WO	32	0x00000000	[31:8] Reserved [7:1] Next transmission data [0] Data direction

寄存器名称	地址偏移	类型	宽度	初始值	描述
RXR	0x0C	RO	32	0x00000000	[31:8] Reserved [7:0] Last received data
CR	0x010	WO	32	0x00000000	[31:8] Reserved [7] STA, Start transmission status [6] STO, Over transmission status [5] RD, Read enable, read data from slave [4] WR, Write enable, write data to slave [3] Acknowledge [2:1] Reserved [0] Interrupt acknowledge
SR	0x14	RO	32	0x00000000	[31:8] Reserved [7] Receive acknowledge signal from slave [6] I2C busy status [5] Arbitration loss [4:2] Reserved [1] Data transmission status flag [0] Interrupt flag

8.3 驱动程序使用方法

I²C Master 驱动程序使用方法，如表 8-2 所示。I²C Master 驱动程序定义位于 library\libraries\drivers\src\GOWIN_M1_i2c.c。

表 8-2 I²C Master 驱动程序使用方法

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_SendData	Send multiple bytes to I2C bus once time
I2C_ReceiveByte	Read a byte from I2C bus
I2C_ReadBytes	Read multiple bytes from I2C bus
I2C_ReceiveData	Read multiple bytes from I2C bus once time
I2C_Rate_Set	Set I2C traffic rate
I2C_Enable	Enable I2C bus
I2C_UnEnable	Disable I2C bus
I2C_InterruptOpen	Open I2C interrupt
I2C_InterruptClose	Close I2C interrupt

9 SPI Master

9.1 特征

Gowin_EMPU_M1，包含一个通过 APB 总线访问的 SPI Master 外设：

- APB 总线接口
- 全双工同步串行数据传输
- 支持 Master 工作模式
- 支持可配置的时钟极性和相位
- SPI 产生的串行时钟频率可配置
- 数据接收寄存器和数据发送寄存器 8 位宽

9.2 寄存器定义

SPI Master 寄存器定义，如表 9-1 所示。SPI Master 寄存器定义位于 library\libraries\cmsis\cm1\device_support\GOWIN_M1.h。

表 9-1 SPI Master 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	32	0x00000000	Read data register [31:8] Reserved [7:0] Read data
WDATA	0x04	WO	32	0x00000000	Write data register [31:8] Reserved [7:0] Write data
STATUS	0x08	RW	32	0x00000000	[31:8] Reserved [7] Overflow error status [6] Receive ready status [5] Transmit ready status [4] Be transmitting [3] Transmit overrun error status [2] Receive overrun error

寄存器名称	地址偏移	类型	宽度	初始值	描述
					status [1:0] Reserved
SSMASK	0x0C	RW	32	0x00000000	[31:1] Reserved [0] Select and enable slave
CTRL	0x10	RW	32	0x00000000	[31:5] Reserved [4:3] Clock selected, CLK_I / 2/4/6/8 [2] Clock polarity [1] Clock phase [0] Direction, 1 is MSB first

9.3 初始化定义

SPI Master 初始化定义，如表 9-2 所示。SPI Master 初始化定义位于 library\libraries\drivers\inc\GOWIN_M1_spi.h。

表 9-2 SPI Master 初始化定义

名称	类型	数值	描述
DIRECTION	uint8_t	1/0	MSB/LSB first transmission 0: MSB first; 1: LSB first.
PHASE	uint8_t	1/0	Posedge/Negedge transmit data 0: Sample at posedge edge; 1: Sample at negedge edge.
POLARITY	uint8_t	1/0	Initialize polarity to one/zero 0: Idle sclk low; 1: Idle sclk high.
CLKSEL	uint8_t	CLKSEL_CLK_DIV_2 CLKSEL_CLK_DIV_4 CLKSEL_CLK_DIV_6 CLKSEL_CLK_DIV_8	Select clock divided 2/4/6/8

9.4 驱动程序使用方法

SPI Master 驱动程序使用方法，如表 9-3 所示。SPI Master 驱动程序定义位于 library\libraries\drivers\src\GOWIN_M1_spi.c。

表 9-3 SPI Master 驱动程序使用方法

名称	描述
SPI_Init	Initializes SPI
SPI_SetDirection	Sets direction
SPI_ClrDirection	Clears direction

名称	描述
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_ReadWriteByte	Full duplex read and write a byte
SPI_WriteData	Writes data
SPI_ReadData	Reads data
SPI_Select_Slave	Select slave

10 RTC

10.1 特征

Gowin_EMPU_M1，包含 1 个通过 APB 总线访问的 32 位 RTC 外设：

- APB 总线接口
- 32-bit 计数器
- 32-bit Match 寄存器
- 32-bit 比较器

MCU 通过 APB 总线接口与 RTC 读写数据、控制和状态信息。在连续输入时钟 CLK1HZ（端口 RTCSRCCLK 接入 3.072MHz 时钟输入，RTC 内部分频为 1Hz）上升沿时，32-bit 计数器递增。

此计数器是不同步计数器，不可重载。在系统复位时，此计数器从 1 开始计数，递增到最大值 0xFFFFFFFF，然后回绕到 0 开始继续递增。

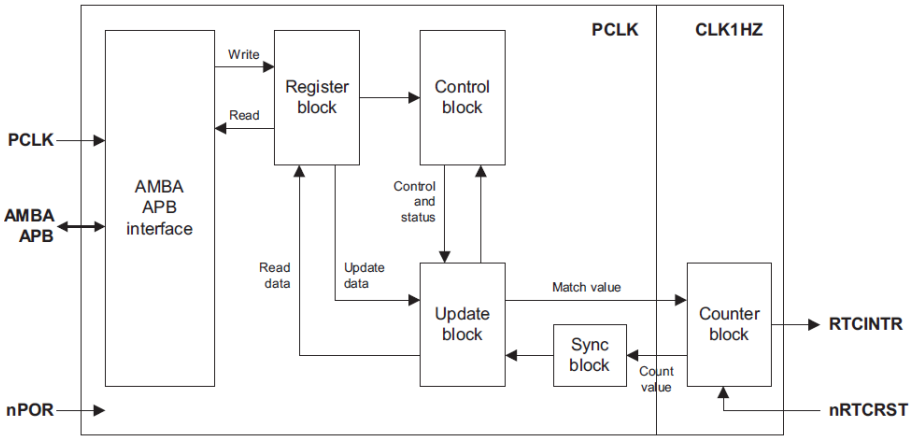
通过写 Load 寄存器 RTC_LOAD_VALUE，实现 RTC 加载或更新。

通过读 Data 寄存器 RTC_CURRENT_DATA，获取 RTC 当前时钟。

通过写 RTC_MATCH_VALUE 寄存器，编程 Match 寄存器。

RTC 结构框图，如图 10-1 所示。

图 10-1 RTC 结构框图



10.2 寄存器定义

RTC 寄存器定义，如表 10-1 所示。RTC 寄存器定义位于 library\libraries\cmsis\cm1\device_support\GOWIN_M1.h。

表 10-1 RTC 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
RTC_CURRENT_DATA	0x000	RO	32	0x00000000	Data Register [31:0] Current value
RTC_MATCH_VALUE	0x004	RW	32	0x00000000	Match Register If current value equals match register's value, generate interrupt. [31:0] Match data
RTC_LOAD_VALUE	0x008	RW	32	0x00000000	Load Register Initialized value, start counter based on this value [31:0] Load data
RTC_CTROLLER_REG	0x00C	RW	32	0x00000000	Control Register Start RTC counter [31:1] Reserved [0] Start RTC counter
RTC_IMSC	0x010	RW	32	0x00000000	Interrupt mask set and clear register Enable or disable interrupt [31:1] Reserved [0] Enable interrupt
RTC_RIS	0x014	RO	32	0x00000000	Raw interrupt status register Get current raw unmasked interrupt status [31:1] Reserved [0] Current raw unmasked interrupt status
RTC_MIS	0x018	RO	32	0x00000000	Masked interrupt status register Get current masked interrupt status [31:1] Reserved [0] Current masked interrupt status
RTC_INTR_CLEAR	0x01C	WO	32	0x00000000	Interrupt clear register Clear current interrupt [31:1] Reserved [0] Clear current

寄存器名称	地址偏移	类型	宽度	初始值	描述
					interrupt

10.3 驱动程序使用方法

RTC 驱动程序使用方法，如表 10-2 所示。RTC 驱动程序定义位于 library\libraries\drivers\src\GOWIN_M1_rtc.c。

表 10-2 RTC 驱动程序使用方法

名称	描述
RTC_init	Initialize RTC
Get_Current_Value	Get RTC current value of data register
Set_Match_Value	Set RTC match value of match register
Get_Match_Value	Get RTC match value of match register
Set_Load_Value	Set RTC load value of load register
Get_Load_Value	Get RTC load value of load register
Start_RTC	Start RTC counter
Close_RTC	Cloase RTC counter
RTC_Inter_Mask_Set	Set RTC interrupt mask
Get_RTC_Control_value	Get value of control register
RTC_Inter_Mask_Clr	Clear RTC interrupt mask
Get_RTC_Inter_Mask_value	Get RTC interrupt mask
Clear_RTC_interrupt	Clear RTC interrupt

11 TRNG

11.1 特征

Gowin_EMPU_M1，包含 1 个通过 APB 总线访问的 32 位 TRNG 外设：

- 数字逻辑产生和采集一个真正的随机数比特流；
- 包含一个基于数字逆变器链的内部熵源；
- 如果 MCU 内核 200MHz 运行，产生一个 10K bits/s 的熵；
- APB 总线接口。

11.2 寄存器定义

TRNG 寄存器定义，如表 11-1 所示。TRNG 寄存器定义位于 library\libraries\cmsis\cm1\device_support\GOWIN_M1.h。

表 11-1 TRNG 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
RESERVE1	0x000-0x0FC	-	-	-	Reserved
RNG_IMR	0x100	RW	32	0x0000000F	Interrupt mask register [31:4] Reserved [3] Mask the Von Neumann error [2] Mask the CRNGT error [1] Mask the Autocorrelation error [0] Mask when the TRNG has collected 192 bits
RNG_ISR	0x104	RO	32	0x00000000	Interrupt status register [31:4] Reserved [3] A Von Neumann error [2] A Continuos Random Number Generation Testing (CRNGT) error [1] The Autocorrelation

寄存器名称	地址偏移	类型	宽度	初始值	描述
					test failed four timers in a row. [0] Set to 1, when 192 bits have been collected, and EHR_DATA[0-5] registers are ready to be read.
RNG_ICR	0x108	WO	32	0x0000 0000	Interrupt clear register [31:4] Reserved [3] Clear a Von Nenumann error [2] Clear a CRNGT error [1] Software cannot clear this bit, only a TRNG reset can clear this bit. [0] Set to 1 after EHR_DATA[0-5] have been read
TRNG_CONFIG	0x10C	RW	32	0x0000 0000	Configuration register [31:2] Reserved [1:0] Selects the number of inverters: 00 = Selects the shortest inverter chain length 01 = Selects the short inverter chain length 10 = Selects the long inverter chain length 11 = Selects the longest inverter chain length
TRNG_VALID	0x110	RO	32	0x0000 0000	Valid register [31:1] Reserved [0] TRNG is complete, data can be read from EHR_DATA[0-5]
EHR_DATA0	0x114—x128	RO	32	0x0000 0000	Entropy holding register data register Return 32 bits from the 192-bit EHR DATA0 returns bit[31:0] DATA1 returns bit[63:32] DATA2 returns bit[95:64] DATA3 returns bit[127:96] DATA4 returns bit[159:128] DATA5 returns bit[191:160]
EHR_DATA1					
EHR_DATA2					
EHR_DATA3					
EHR_DATA4					
EHR_DATA5					
RND_SOURCE_ENABLE	0x12C	RW	32	0x0000 0000	Random source enable register

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[31:1] Reserved [0] 1 = enable entropy source; 0 = disable entropy source
SAMPLE_CNT1	0x130	RW	32	0x0000 FFFF	Sample count register [31:0] Sets the number of rng_clk cycles
AUTOCORR_STATISTIC	0x134	RW	32	0x0000 0000	Autocorrelation register [31:22] Reserved [21:14] Count each time an autocorrelation test fails [13:0] Count each time an autocorrelation test starts
TRNG_DEBUG_CONTROL	0x138	RO	32	0x0000 0000	Debug control register [31:4] Reserved [3] The autocorrelation test is bypassed [2] The CRNGT test is bypassed [1] The Von Neumann balancer is bypassed [0] Reserved
RESERVE2	0x13C	-	-	-	Reserved
TRNG_SW_RESET	0x140	WO	32	0x0000 0000	Reset register [31:1] Reserved [0] Writing 1 to this register causes an internal TRNG reset
RESERVE3	0x144-0x1B4	-	-	-	Reserved
TRNG_BUSY	0x1B8	RO	32	0x0000 0000	Busy register [31:1] Reserved [0] Relects the status of rng_busy signal
RST_BIT_COUNT	0x1BC	WO	32	0x0000 0000	Reset bit counter register [31:1] Reserved [0] Write any value to this bit resets the bits counter and TRNG valid registers
RESERVE4	0x1C0-0x1DC	-	-	-	Reserved
RNG_BIST_CNTR0	0x1E0-0x1E8	RO	32	0x0000 0000	BIST counter registers Return the collected BIST results [31:22] Reserved [21:0] Returns the results of the TRNG BIST
RNG_BIST_CNTR1					
RNG_BIST_					

寄存器名称	地址偏移	类型	宽度	初始值	描述
CNTR2					counter

11.3 驱动程序使用方法

TRNG 驱动程序使用方法，如表 11-2 所示。TRNG 驱动程序定义位于 library\libraries\drivers\src\GOWIN_M1_trng.c。

表 11-2 TRNG 驱动程序使用方法

名称	描述
Init_TRNG	Initialized TRNG
Set_Interrupt_Mask	Set interrupt mask
Get_Int_State	Get interrupt status
Clear_Int	Clear interrupt
Set_Config	Set config register
Get_EHR_Data	Get Entropy holding data
Set_Random_Source_Enable	Set random source enable register
Clr_Random_Source_Enable	Clear random source enable register
Set_Sample_Count	Set sample count register
Trng_SW_Reset	Reset TRNG
Get_TRNG_State	Get TRNG state
Reset_Bit_Count	Reset bit count register
Get_BIT_Counter	Get bits count register
Set_Debug_Control	Set debug control register
Fail_Start_State_times	Get autocorrelation register
Clr_Fail_Start_State_register	Clear autocorrelation register

12 DualTimer

12.1 特征

Gowin_EMPU_M1，包含 1 个通过 APB 总线访问的 32 位和 16 位 DualTimer 外设：

- APB 总线接口
- 包含两个可编程的 32-bit 或 16-bit 倒数器
- 倒计数器倒数到 0 时能够产生中断

12.2 寄存器定义

DualTimer 寄存器定义，如表 12-1 所示。DualTimer 寄存器定义位于 library\libraries\cmsis\cm1\device_support\GOWIN_M1.h。

表 12-1 DualTimer 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
TIMER1LOAD	0x00	RW	32	0x0000 0000	Timer1 load register [31:0] Timer1 load value
TIMER1VALUE	0x04	RO	32	0xFFFF FFFF	Timer1 current value register [31:0] Timer1 current value
TIMER1CONTROL	0x08	RW	32	0x0000 0020	Timer1 control register [31:8] Reserved [7] Timer enable [6] Timer mode [5] Interrupt enable [4] Reserved [3:2] Timer prescale 00 = clock is divided by 1 01 = clock is divided by 16 10 = clock is divided by 256 11 Undefined [1] Timer size 0 = 16-bit counter, default 1 = 32-bit counter

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[0] One-shot count 0 = wrapping mode, default 1 = on-shot mode
TIMER1INTCLR	0x0C	WO	-	-	Timer1 interrupt clear register Any write clears the interrupt output of the counter
TIMER1RIS	0x10	RO	32	0x0000 0000	Timer1 raw interrupt status register [31:1] Reserved [0] Raw interrupt status from the counter
TIMER1MIS	0x14	RO	32	0x0000 0000	Timer1 interrupt status register [31:1] Reserved [0] Enable interrupt status from the counter
TIMER1BGLOAD	0x18	RW	32	0x0000 0000	Timer1 background load register [31:0] The value used to reload the counter
RESERVE1	-	-	-	-	Reserved
TIMER2LOAD	0x00	RW	32	0x0000 0000	Timer2 load register [31:0] Timer2 load value
TIMER2VALUE	0x04	RO	32	0xFFFF FFFF	Timer2 current value register [31:0] Timer2 current value
TIMER2CONTROL	0x08	RW	32	0x0000 0020	Timer2 control register [31:8] Reserved [7] Timer enable [6] Timer mode [5] Interrupt enable [4] Reserved [3:2] Timer prescale 00 = clock is divided by 1 01 = clock is divided by 16 10 = clock is divided by 256 11 Undefined [1] Timer size 0 = 16-bit counter, default 1 = 32-bit counter [0] One-shot count 0 = wrapping mode, default 1 = on-shot mode
TIMER2INTCLR	0x0C	WO	-	-	Timer2 interrupt clear register Any write clears the

寄存器名称	地址偏移	类型	宽度	初始值	描述
					interrupt output of the counter
TIMER2RIS	0x10	RO	32	0x0000 0000	Timer2 raw interrupt status register [31:1] Reserved [0] Raw interrupt status from the counter
TIMER2MIS	0x14	RO	32	0x0000 0000	Timer2 interrupt status register [31:1] Reserved [0] Enable interrupt status from the counter
TIMER2BGLOAD	0x18	RW	32	0x0000 0000	Timer2 background load register [31:0] The value used to reload the counter

12.3 驱动程序使用方法

DualTimer 驱动程序使用方法，如表 12-2 所示。DualTimer 驱动程序定义位于 library\libraries\drivers\src\GOWIN_M1_dualtimer.c。

表 12-2 DualTimer 驱动程序使用方法

名称	描述
DUALTIMER1_Init	Initialized DualTimer1
DUALTIMER2_Init	Intiialized DualTimer2
Clear_DULATIMER_interrupt	Clear DualTimer interrupt
Dtimer_MODE_function	Set timer mode of DualTimer1 or DualTimer2
Dtimer_PRE_function	Set timer prescale of DualTimer1 or DualTimer2
INIT_NUM_load_function	Set load value of DualTimer1 or DualTimer2
ENABLE_interrupt_Dtimer_function	Enable interrupt of DualTimer1 or DualTimer2
TIMER_SIZE_function	Set timer size of DualTimer1 or DualTimer2
ENABLE_Dtimer_function	Enable DualTimer1 or DualTime2
Get_DULATIMER_interrupt_num	Get timer ID of DualTimer1 or DualTimer2

13 SD-Card

13.1 特征

Gowin_EMPU_M1，包含 1 个通过 APB 总线访问的 SD-Card 外设：

- 支持 SD/MMC 格式的存储卡
- 支持存储卡硬件初始化
- 简单的 SPI 总线访问方式
- 支持 block 读和 block 写
- 内置 512 字节的接收和发送缓存区
- APB 总线接口
- APB 接口和 SPI 内核逻辑独立时钟
- 数据传输速度接近 SD/MMC 存储卡的最大速率

13.2 寄存器定义

SD-Card 寄存器定义，如表 13-1 所示。SD-Card 寄存器定义位于 library\libraries\cmsis\cm1\device_support\GOWIN_M1.h。

表 13-1 SD-Card 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
SPI_MASTER_VERSION	0x000	RW	8	0x00	SPI master version register [7:4] Major revision number [3:0] Minor revision number
SPI_MASTER_CONTROL	0x001	WO	8	0x00	SPI master control register [7:1] Reserved [0] Reset core logic and register 1 = Reset core logic and register, self clearing
TRANS_TYPE	0x002	RW	8	0x00	Transaction type register [7:2] Reserved [1:0] Set the transaction type

寄存器名称	地址偏移	类型	宽度	初始值	描述
					00 = Direct access 01 = Initialized SD 10 = Read SD block 11 = Write SD block
TRANS_CTRL	0x003	WO	8	0x00	Transaction control register [7:1] Reserved [0] Start transaction 1 = Start transaction, self clearing
TRANS_STS	0x004	RO	8	-	Transaction status register [7:1] Reserved [0] Transaction busy 1 = Transaction busy
TRANS_ERROR	0x005	RO	8	-	Transaction error register [7:6] Reserved [5:4] SD write error 00 = Write no error 01 = Write command error 10 = Write data error 11 = Write busy error [3:2] SD read error 00 = Read no error 01 = Read command error 10 = Read token error [1:0] SD initialize error 00 = Initialize no error 01 = Initialize command 0 error 10 = Initialize command 1 error
DIRECT_ACCESS_DATA	0x006	RW	8	0x00 / -	Data direct access register [7:0] Transmit data Set TX_DATA prior to starting a DIRECT_ACCESS transaction [7:0] Receive data Read RX_DATA after completing a DIRECT_ACCESS transaction
SD_ADDR_7_0	0x007	RW	8	0x00	SD address [7:0] bit register [7:0] SD_ADDR[7:0]
SD_ADDR_15_8	0x008	RW	8	0x00	SD address [15:8] bit register [7:0] SD_ADDR[15:8]
SD_ADDR_23_1	0x009	RW	8	0x00	SD address [23:16] bit

寄存器名称	地址偏移	类型	宽度	初始值	描述
6					register [7:0] SD_ADDR[23:16]
SD_ADDR_31_24	0x00A	RW	8	0x00	SD address [31:24] bit register [7:0] SD_ADDR[31:24]
SPI_CLK_DEL	0x00B	RW	8	0x00	SPI clock control register [7:0] Control the frequency of the SPI_CLK after SD initialization is completed
RESERVED0	0x00C-0x00F	-	-	-	Reserved
RX_FIFO_DATA	0x010	RW	8	-	SPI block reading data register [7:0] SD/MMC block read data, fifo size matches the SD/MMC block size of 512 bytes
RESERVED1	0x011	-	-	-	Reserved
RX_FIFO_DATA_COUNT_MSB	0x012	RO	8	-	MSB byte of reading data count register [7:0] MSByte of FIFO_DATA_COUNT, indicates the number of data entries within the fifo
RX_FIFO_DATA_COUNT_LSB	0x013	RO	8	-	LSB byte of reading data count register [7:0] LSByte of FIFO_DATA_COUNT, indicates the number of data entries within the fifo
RX_FIFO_CONTROL	0x014	WO	8	0x00	SD block reading data control register [7:1] Reserved [0] Force fifo empty 1 = Force fifo empty, delete all the data samples within the fifo, self clearing
RESERVED2	0x015-0x019	-	-	-	Reserved
TX_FIFO_DATA	0x020	WO	8	-	SD block writing data register [7:0] SD/MMC block write data, fifo size matches the SD/MMC block size of 512 bytes
RESERVED3	0x021-0x023	-	-	-	Reserved
TX_FIFO_CONTROL	0x024	WO	8	0x00	SD block writing data control register

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[7:1] Reserved [0] Force fifo empty 1 = Force fifo empty, delete all the data samples within the fifo, self clearing

13.3 驱动程序使用方法

SD-Card 驱动程序使用方法，如表 13-2 所示。SD-Card 驱动程序定义位于 library\libraries\drivers\src\GOWIN_M1_sdcard.c。

表 13-2 SD-Card 驱动程序使用方法

名称	描述
SD_Init	SD hardware initialization
SD_BlockWrite	SD block write data, block size is 512 bytes.
SD_BlockRead	SD block read data, block size is 512 bytes.

14 CAN

14.1 特征

Gowin_EMPU_M1，包含 1 个通过 AHB 总线访问的 CAN 外设：

- AHB 总线接口
- 符合 CAN2.0A 和 CAN2.0B 协议及 ISO 11898-1 标准
- CAN FD 协议
- 独立系统时钟和 CAN 总线时钟
- 灵活的共享缓存方案，实现最佳缓存区大小，以便在给定的应用程序中存储发送和接收消息
- 接收滤波器可配置为 1-16 个
- 可编程波特率预分频器

14.2 寄存器定义

CAN 寄存器定义，如表 14-1 所示。CAN 寄存器定义位于 library\libraries\cmsis\cm1\device_support\GOWIN_M1.h。

表 14-1 CAN 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
SRST	0x0000	RW	32	0x00000000	Software reset register [31:1] Reserved [0] Control reset 1 = Hardware reset 0 = Software reset
CMD	0x0004	RW	32	0x00000000	Command register [31:1] Reserved [0] Command settings 1 = Working mode 0 = Command mode
BRP	0x0008	RW	32	0x00000000	Baud rate prescalar register [31:8] Reserved

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[7:0] Baud rate prescalar
BTN	0x000C	RW	32	0x00000000	Bit timing (nominal) register [28:24] sjw_nom [13:8] phseg2_nom, PHASE_SEG2 window's width [5:0] phseg1_nom, PROP_SEG+PHASE_SEG1 window's width
BTD	0x0010	RW	32	0x00000000	Bit timing (data) register [26:24] sjw_d [11:8] phseg2_d, PHASE_SEG2 window's depth [3:0] phseg1_d, PROP_SEG+PHASE_SEG1 window's depth
RSVD0	0x001C	-	-	-	Reserved
IS	0x0020	RO	32	0x00000000	Interrupt status register [31] Bus off status [27] TX message successfully [26] TX message retry [25] TX message failed [23] TX high-priority message successfully [22] TX high-priority message retry [21] TX high-priority message failed [8] Error status [5] TX high-priority fifo overflow [4] TX fifo overflow [1] RX fifo overflow [0] RX fifo valid
IE	0x0024	RW	32	0x00000000	Interrupt enable register [31] Enable us off [27] Enable TX message successfully [26] Enable TX message retry [25] Enable TX message failed [23] Enable TX high-priority message successfully [22] Enabe TX high-

寄存器名称	地址偏移	类型	宽度	初始值	描述
					priority message retry [21] Enable TX high-priority message failed [8] Enable Error status [5] Enabe TX high-priority fifo overflow [4] Enable TX fifo overflow [1] Enable RX fifo overflow [0] Enable RX fifo valid
IC	0x0028	WO	32	0x00000000	Interrupt clear register [31] Clear bus off status [27] Clear TX message successfully status [26] Clear TX message retry status [25] Clear TX message failed status [23] Clear TX high-priority message successfully status [22] Clear TX high-priority message retry status [21] Clear TX high-priority message failed status [8] Clear Error status status [5] Clear TX high-priority fifo overflow status [4] Clear TX fifo overflow status [1] Clear RX fifo overflow status [0] Clear RX fifo valid status
RSVD1	0x002C	-	-	-	Reserved
CFG	0x0030	RW	32	0x00000000	Configuration register [4] Configure disprotexceponres 1 = ‘res’ is FORM-ERROR 0 = ‘res’ is exception [0] Configure isofd 1 = ISO FD mode 0 = non ISO FD mode
RSVD2	0x0034-	-	-	-	Reserved

寄存器名称	地址偏移	类型	宽度	初始值	描述
	0x003C				
RXBCFG	0x0040	RW	32	0x00000000	RX buffer/fifo configuration register [31:16] RX buffer's ending offset [15:0] RX buffer's start offset
TXBCFG	0x0044	RW	32	0x00000000	TX buffer/fifo configuration register [31:16] TX buffer's ending offset [15:0] TX buffer's start offset
TXHBCFG	0x0048	RW	32	0x00000000	TX high-priority/fifo configuration register [31:16] TX high-priority buffer's ending offset [15:0] TX high-priority buffer's start offset
RSVD3	0x004C	-	-	-	Reserved
TXBRETRY	0x0050	RW	32	0x00000000	TX buffer retry counter
TXHBRETRY	0x0054	RW	32	0x00000000	TX high-priority buffer retry counter
TXMSGSTS	0x0058	RO	32	0x00000000	Transmit message status register [31:30] TX message status 00 = Successfully 10 = Retry 11 = Failed [28:0] Message ID
TXHMSGSTS	0x005C	RO	32	0x00000000	Transmit high-priority message status register [31:30] TX high-priority message status 00 = Successfully 10 = Retry 11 = Failed [28:0] Message ID
ERRSTS	0x0060	RW	32	0x00000000	Error status register [4] CRC error [3] ACK error [2] FORM error [1] BIT error [0] STUFF error
ERRCNTR	0x0064	RO	32	0x00000000	Error counter register [24:16] TX error counter [8:0] RX error counter

寄存器名称	地址偏移	类型	宽度	初始值	描述
RSVD4	0x0068-0x00FC	-	-	-	Reserved
AF	0x0100-0x100+(4*N)	RW	32	0x00000000	Receive acceptance filter register [31] Enable 1 = Valid 0 = Invalid [30] IDE 1 = Extended frame 0 = Normal frame [29] Extended data length 1 = Match FD frame 0 = Match normal frame [28:18] Basic ID [17:0] ID Extension
AFM	0x0140-0x140+(4*N)	RW	32	0x00000000	Receive acceptance filter mask register [28:18] Basic ID mask [17:0] ID Extension mask
RSVD5	0x0180-0x01FC	-	-	-	Reserved
RXB	0x0200	RO	32	0x00000000	Receive buffer/fifo window register
TXB	0x0204	WO	32	0x00000000	Transmit buffer/fifo window register
TXHB	0x0208	WO	32	0x00000000	Transmit high-priority buffer/fifo window register
TXBSTS	0x020C	RO	32	0x00000000	Transmit buffer/fifo status [31] txbwerr [15:0] txbspace
TXHBSTS	0x0210	RO	32	0x00000000	Transmit high-priority buffer/fifo status [31] txhbwerr [15:0] txhbspace
RXBSTS	0x0214	RO	32	0x00000000	Receive buffer/fifo status [15:0] rxbdepth

14.3 驱动程序使用方法

CAN 驱动程序使用方法，如表 14-2 所示。CAN 驱动程序定义位于 library\libraries\drivers\src\GOWIN_M1_can.c。

表 14-2 CAN 驱动程序使用方法

名称	描述
can_srst	Start hard reset
can_set_cmd	Enable working mode
can_set_brp	Set baud rate prescalar
can_set_btn_phseg1_nom	Set PROP_SEG+PHASE_SEG1 window's width
can_set_btn_phseg2_nom	Set PHASE_SEG2 window's width
can_set_btn_sjw_nom	Set sjw_nom
can_set_btn	Set BTN register
can_read_btn_phseg1_nom	Get PROP_SEG+PHASE_SEG1 window's width
can_read_btn_phseg2_nom	Get PHASE_SEG2 window's width
can_read_btn_sjw_nom	Get sjw_nom
can_set_btd_phseg1_d	Set PROP_SEG+PHASE_SEG1 window's depth
can_set_btd_phseg2_d	Set PHASE_SEG2 window's depth
can_set_btd_sjw_d	Set sjw_d
can_set_btd	Set BTD register
can_read_btd_phseg1_d	Get PROP_SEG+PHASE_SEG1 window's depth
can_read_btd_phseg2_d	Get PHASE_SEG2 window's depth
can_read_btd_sjw_d	Get sjw_d
can_read_is_bit	Get IS register bits function
can_set_ie_bit	Set IE register bits function
can_clear_ie_bit	Clear IE register bits function
can_read_ie_bit	Get IE register bits function
can_set_ic_bit	Set IC register bits function
can_set_cfg_bit_as_one	Set CFG register bits function
can_set_cfg_bit_as_zero	Clear CFG register bits function
can_read_cfg_bit	Get CFG register bits function
can_set_rxbcfg_rxb_start	Set RX buffer start offset in RXBCFG
can_read_rxbcfg_rxb_start	Get RX buffer start offset in RXBCFG
can_set_rxbcfg_rxb_end	Set RX buffer ending offset in RXBCFG
can_read_rxbcfg_rxb_end	Get RX buffer ending offset in RXBCFG
set_rxbcfg	Set RX buffer start and ending offset
can_set_txbcfg_txb_start	Set TX buffer start offset in TXBCFG
can_read_txbcfg_txb_start	Get TX buffer start offset in TXBCFG
can_set_txbcfg_txb_end	Set TX buffer ending offset in TXBCFG
can_read_txbcfg_txb_end	Get TX buffer ending offset in TXBCFG
set_txbcfg	Set TX buffer start and ending offset

名称	描述
can_set_txhbcfg_txhb_start	Set TX high-priority buffer start offset in TXHBCFG
can_read_txhbcfg_txhb_start	Get TX high-priority buffer start offset in TXHBCFG
can_set_txhbcfg_txhb_end	Set TX high-priority buffer ending offset in TXHBCFG
can_read_txhbcfg_txhb_end	Get TX high-priority buffer ending offset in TXHBCFG
set_txhbcfg	Set TX high-priority buffer start and ending offset
can_set_txbretry	Set TX buffer retry
can_read_txbretry	Get TX buffer retry
can_set_txhbretry	Set TX high-priority buffer retry counter
can_read_txhbretry	Get TX high-priority buffer retry counter
can_read_txmsgsts	Get TX message status
can_read_txmsgid	Get TX message ID
can_read_txhmsgsts	Get TX high-priority message status
can_read_txhmsgid	Get TX high-priority message ID
can_read_errsts	Get error status
can_read_errcntr_rec	Get RX error counter
can_read_errcntr_tec	Get TX error counter
can_set_af_bit_as_one	Set AF bits function
can_set_af_bit_as_zero	Clear AF bits function
can_read_af_bit	Get AF bits function
can_set_af_ie	Set AF ID Extension
can_read_af_ie	Get AF ID Extension
can_set_af_bid	Set AF basic ID
can_read_af_bid	Get AF basic ID
can_set_afm_ie	Set AFM ID Extension mask
can_read_afm_ie	Get AFM ID Extension mask
can_set_afm_bid	Set AFM basic ID mask
can_read_afm_bid	Get AFM basic ID mask
can_read_rxb	Get RXB register
can_set_txb	Set TXB register
can_set_txhb	Set TXHB register
can_read_txbsts_tdbspace	Get txbspace
can_read_txbsts_txbwerr	Get txbwerr
can_read_txhbsts_txhbspace	Get txhbspace
can_read_txhbsts_txhbwerr	Get txhbwerr
can_read_rxbsts	Get rxbdepth

15 Ethernet

15.1 特征

Gowin_EMPU_M1，包含 1 个通过 AHB 总线访问的 Ethernet 外设：

- AHB 总线接口
- 实现 IEEE802.3 协议中对 Ethernet MAC 层的功能描述
- 支持 RGMII/GMII/MII 接口
- 支持 10/100/1000M 速率
- 支持全双工和半双工模式，半双工模式下支持冲突检测
- 支持用户可选是否自动添加和校验 CRC
- 支持自动添加 pad 功能
- 支持以太网帧分类统计
- 支持以太网帧错误统计
- 支持 IFG 可配置功能
- 支持 Jumbo 模式
- 支持全双工模式下的 Flow Control
- 支持 Management 接口 mdc、mdio

15.2 寄存器定义

Ethernet 寄存器定义，如表 15-1 所示。Ethernet 寄存器定义位于 library\libraries\cmsis\cm1\device_support\GOWIN_M1.h。

表 15-1 Ethernet 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
ETH_TX_DATA	0x000-0x5FF	WO	32	0x00000000	Transmit data registers
ETH_RX_DATA	0x000-0x5FFF	RO	32	0x00000000	Receive data registers
ETH_TX_LENGTH	0x600	RW	32	0x00000000	Transmit data length [31:11] Reserved [10:0] TX data length

寄存器名称	地址偏移	类型	宽度	初始值	描述
ETH_TX_EN	0x604	RW	32	0x00000000	Transmit enable [31:1] Reserved [0] Enable TX
ETH_TX_FAIL	0x608	RW	32	0x00000000	Transmit failed status [31:3] Reserved [2] TX late [1] TX excessive [0] TX failed
ETH_TX_IS	0x60C	RO	32	0x00000000	Transmit interrupt status [31:1] Reserved [0] TX interrupt status
ETH_TX_IC	0x610	WO	32	0x00000000	Transmit interrupt clear [31:1] Reserved [0] Clear TX interrupt
ETH_TX_IE	0x614	RW	32	0x00000000	Transmit interrupt enable [31:1] Reserved [0] Enable TX interrupt
RESERVED_1	0x618-0x67F	-	-	-	Reserved
ETH_RX_LEGHT	0x680	RO	32	0x00000000	Receive data length
ETH_RX_IS	0x684	RO	32	0x00000000	Receive interrupt status [31:1] Reserved [0] RX interrupt status
ETH_RX_IE	0x688	RW	32	0x00000000	Receive interrupt enable [31:1] Reserved [0] Enable RX interrupt
ETH_RX_IC	0x68C	WO	32	0x00000000	Receive interrupt clear [31:1] Reserved [0] Clear RX interrupt
RESERVED_2	0x690-0x6FFF	-	-	-	Reserved
MIIM_OP_MODE	0x700	RW	32	0x00000000	MIIM operation mode [31:1] Reserved [0] MIIM operation mode
MIIM_PHY_ADDR	0x704	RW	32	0x00000000	MIIM PHY address [31:5] Reserved [4:0] MIIM PHY address
MIIM_REG_ADDR	0x708	RW	32	0x00000000	MIIM reg address [31:5] Reserved [4:0] MIIM reg address
MIIM_WR_DATA	0x70C	RW	32	0x00000000	MIIM write data [31:16] Reserved [15:0] MIIM write data
MIIM_RD_DATA	0x710	RO	32	0x00000000	MIIM read data [31:16] Reserved [15:0] MIIM read data

寄存器名称	地址偏移	类型	宽度	初始值	描述
MIIM_IS	0x714	RO	32	0x00000000	MIIM interrupt status [31:2] Reserved [1] MIIM operation end [0] MIIM read data valid
MIIM_IE	0x718	RW	32	0x00000000	MIIM interrupt enable [31:2] Reserved [1] MIIM operation end [0] MIIM read data valid
MIIM_IC	0x71C	WO	32	0x00000000	MIIM interrupt clear [31:2] Reserved [1] MIIM operation end [0] MIIM read data valid
MIIM_OP_EN	0x720	RW	32	0x00000000	MIIM operation enable [31:1] Reserved [0] Enable MIIM operation
ETH_MODE	0x724	RW	32	0x00000000	Ethernet operation mode [31:3] Reserved [2:0] Duplex mode and speed 000 = Full duplex 100M 001 = Full duplex 1000M 010 = Full duplex 10M 100 = Half duplex 100M 110 = Half duplex 10M

15.3 驱动程序使用方法

Ethernet 驱动程序使用方法，如表 15-2 所示。Ethernet 驱动程序定义位于 library\libraries\drivers\src\GOWIN_M1_ethernet.c。

表 15-2 Ethernet 驱动程序使用方法

名称	描述
eth_init	Initialize Ethernet
tx_int_event	TX interrupt
rx_int_event	RX interrupt
eth_tx	Ethernet TX
eth_set_mode	Set Ethernet duplex mode and speed
miim_wr_int_event	MIIM interface transmits interrupt
miim_rd_int_event	MIIM interface receives interrupt
miim_write	MIIM interface transmits data
miim_receive	MIIM interface receives data

16 DDR3 Memory

16.1 特征

Gowin_EMPU_M1，包含 1 个通过 AHB 总线访问的 DDR3 Memory 外设：

- AHB 总线接口
- 能与工业标准的 DDR3 SDRAM 器件和具有 JESD79-3F 规范兼容的模块接口
- 支持存储器数据路径宽度为 16 位
- 支持 UDIMM 内存模块
- 支持 x8 数据宽度的内存芯片
- 可编程突发长度 4
- 支持时钟比例 1:2 模式

16.2 寄存器定义

DDR3 Memory 寄存器定义，如表 16-1 所示。DDR3 Memory 寄存器定义位于 library\libraries\cmsis\cm1\device_support\GOWIN_M1.h。

表 16-1 DDR3 Memory 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
RESERVED	0x0000	-	-	-	Reserved
WR_ADDR	0x0004	RW	32	0x0	Write address register
WR_DATA	0x0008-0x0014	WO	128	0x0	Write data register
RD_ADDR	0x0018	RW	32	0x0	Read address register
RD_EN	0x001c	RW	32	0x0	Read enable register [31:1] Reserved [0] Read enable 1 = Enable 0 = Disable
RD_DATA	0x0020-0x002c	RO	128	0x0	Read data register

寄存器名称	地址偏移	类型	宽度	初始值	描述
INIT	0x0030	RW	32	0x0	Initialized completely flag register [31:1] Reserved [0] Initialzed completely flag
WR_EN	0x0034	RW	32	0x0	Write enable and ending flag register [31:1] Reserved [0] Write enable and ending flag 1 = Enable 0 = Ending

16.3 驱动程序使用方法

DDR3 Memory 驱动程序使用方法，如表 16-2 所示。DDR3 Memory 驱动程序定义位于 library\libraries\drivers\src\GOWIN_M1_ddr3.c。

表 16-2 DDR3 Memory 驱动程序使用方法

名称	描述
DDR3_Init	Initialize DDR3 Memory
DDR3_Read	Read data from DDR3 Memory
DDR3_Write	Write data into DDR3 Memory

17 SPI-Flash Memory

17.1 特征

Gowin_EMPU_M1，包含 1 个通过 AHB 总线访问的 SPI-Flash Memory 外设：

- SPI-Flash Memory 下载功能为 AHB 总线接口
- SPI-Flash Memory 读、写和擦除功能为 APB 总线接口
- 支持 Quad SPI-Flash Memory 读、写和擦除功能

17.2 寄存器定义

SPI-Flash Memory 寄存器定义，如表 17-1 所示。SPI-Flash Memory 寄存器定义位于 library\libraries\cmsis\cm1\device_support\GOWIN_M1.h。

表 17-1 SPI-Flash Memory 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
IDREV	0x00	RO	32	0x02002000	ID and revision register [31:8] ID number [7:4] Major revision number [3:0] Minor revision number
RESERVED0[3]	0x04-0x0C	-	-	-	Reserved
TRANSFMT	0x10	RW	32	0x00020780	SPI transfer format register [31:18] Reserved [17:16] Address length in bytes 00 = 1 byte 01 = 2 bytes 10 = 3 bytes 11 = 4 bytes [15:13] Reserved [12:8] Data length [7] Enable data merge mode [6:5] Reserved [4] Bi-directional MOSI in single mode 0 = MOSI is uni-directional signal

寄存器名称	地址偏移	类型	宽度	初始值	描述
					1 = MOSI is bi-directional signal [3] Transfer data with the lease significant bit first 0 = Most significant bit first 1 = Least significant bit first [2] SPI master/slave mode selection 0 = Master mode 1 = Slave mode [1] SPI clock polarity 0 = SCLK is LOW in the idle states 1 = SCLK is HIGH in the idle states [0] SPI clock phase 0 = Sampling data at odd SCLK edges 1 = Sampling data at even SCLK edges
DIRECTIO	0x14	RW	32	0x0	SPI direct IO control register [31:25] Reserved [24] Enable direct IO 0 = Disable 1 = Enable [23:22] Reserved [21] Output enable for SPI-Flash hold signal [20] Output enable for SPI-Flash write protect signal [19] Output enable for the SPI MISO signal [18] Output enable for the SPI MOSI signal [17] Output enable for SPI SCLK signal [16] Output enable for SPI CS signal [15:14] Reserved [13] Output value for SPI-Flash hold signal [12] Output value for SPI-Flash write protect signal [11] Output value for SPI MISO signal [10] Output value for SPI MOSI signal [9] Output value for SPI SCLK signal [8] Output value for SPI CS signal [7:6] Reserved [5] Status of SPI-Flash hold signal [4] Status of SPI-Flash write protect signal [3] Status of SPI MISO signal [2] Status of SPI MOSI signal [1] Status of SPI SCLK signal [0] Status of SPI CS signal

寄存器名称	地址偏移	类型	宽度	初始值	描述
RESERVED1[2]	0x18-0x1C	-	-	-	Reserved
TRANSCTRL	0x20	RW	32	0x0	SPI transfer control register [31] Reserved [30] SPI command phase enable 0 = Disable the command phase 1 = Enable the command phase (Master mode only) [29] SPI address phase enable 0 = Disable the address phase 1 = Enable the address phase (Master mode only) [28] SPI address phase format 0 = Address phase is single mode 1 = The format of the address phase is the same as the DualQuad data phase (Master mode only) [27:24] Transfer mode 0000 = Write and read at the same time 0001 = Write only 0010 = Read only 0011 = Write, Read 0100 = Read, Write 0101 = Write, Dummy, Read 0110 = Read, Dummy, Write 0111 = None data 1000 = Dummy, Write 1001 = Dummy, Read 1010~1111 = Reserved [23:22] SPI data phase format 00 = Single mode 01 = Dual I/O mode 10 = Quad I/O mode 11 = Reserved [21] Append and one-byte special token following the address phase for SPI read transfers [20:12] Transfer count for write data [11] The value of the one-byte special token following the address phase for SPI read transfers 0 = token value is 0x00 1 = token value is 0x69 [10:9] Dummy data count [8:0] Transfer count for read data
CMD	0x24	RW	32	0x0	SPI command register [31:8] Reserved

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[7:0] SPI command
ADDR	0x28	RW	32	0x0	SPI address register [31:0] SPI address (Master mode only)
DATA	0x2C	RW	32	0x0	SPI data register [31:0] Data to transmit or the received data
CTRL	0x30	RW	32	0x0	SPI controller register [31:21] Reserved [20:16] Transmit FIFO threshold [15:13] Reserved [12:8] Receive FIFO threshold [7:5] Reserved [4] TX DMA enable [3] RX DMA enable [2] Transmit FIFO reset [1] Receive FIFO reset [0] SPI reset
STATUS	0x34	RO	32	0x0	SPI status register [31:24] Reserved [23] Transmit FIFO full flag [22] Transmit FIFO empty flag [21] Reserved [20:16] Number of valid entries int the transmit FIFO [15] Receive FIFO full flag [14] Receive FIFO empty flag [13] Reserved [12:8] Number of valid entries in the receive FIFO [7:1] Reserved [0] SPI register programming is in progress
INTREN	0x38	RW	32	0x0	SPI interrupt enable register [31:6] Reserved [5] Enable the slave command interrupt [4] Enable the end of SPI transfer interrupt [3] Enable the SPI transmit FIFO threshold interrupt [2] Enable the SPI receive FIFO threshold interrupt [1] Enable SPI transmit FIFO underrun interrupt (Slave mode only) [0] Enable SPI receive FIFO overrun interrupt (Slave mode only)
INTRST	0x3C	WO	32	0x0	SPI interrupt status register [31:6] Reserved

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[5] Slave command interrupt (Slave mode only) [4] End of SPI transfer interrupt [3] TX FIFO threshold interrupt [2] RX FIFO threshold interrupt [1] TX FIFO underrun interrupt (Slave mode only) [0] RX FIFO overrun interrupt (Slave mode only)
TIMING	0x40	RW	32	0x0	SPI interface timing register [31:14] Reserved [13:12] The minimum time between the edges of SPI CS and the edges of SCLK [11:8] The minimum time the SPI CS should stay HIGH [7:0] The clock frequency ratio between the clock source and SPI interface SCLK
RESERVED2[3]	0x44-0x4C	-	-	-	Reserved
MEMCTRL	0x50	RW	32	0x0	SPI memory access control register [31:9] Reserved [8] This bit is set when “MEMCTRL” / “TIMING” is written [7:4] Reserved [3:0] Selects the SPI command
RESERVED3[3]	0x54-0x5C	-	-	-	Reserved
SLVST	0x60	RW	32	0x0	SPI slave status register [31:19] Reserved [18] Data underrun occurs in the last transaction [17] Data overrun occurs in the last transaction [16] SPI is ready for data transaction [15:0] User defined status flags
SLVDATACNT	0x64	RO	32	0x0	SPI slave data count register [31:25] Reserved [24:16] Slave transmitted data count [15:9] Reserved [8:0] Slave received data count
RESERVED4[5]	0x68-0x78	-	-	-	Reserved
CONFIG	0x7C	RO	32	0x0	Configuration register [31:15] Reserved [14] Support for SPI slave mode [13] Reserved [12] Support for memory-mapped access through AHB bus [11] Support for direct SPI IO

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[10] Reserved [9] Support for Quad I/O SPI [8] Support for Dual I/O SPI [7:6] Reserved [5:4] Depth of TX FIFO 00 = 2 words 01 = 4 words 10 = 8 words 11 = 16 words [3:2] Reserved [1:0] Depth of RX FIFO 00 = 2 words 01 = 4 words 10 = 8 words 11 = 16 words

17.3 驱动程序使用方法

~~17.2.1~~17.3.1 QSPI-Flash 驱动程序

QSPI-Flash Memory 驱动程序使用方法如表 17-2 所示。QSPI-Flash Memory 驱动程序定义位于 library\libraries\drivers\src\GOWIN_M1_qspi_flash.c。

表 17-2 QSPI-Flash Memory 驱动程序使用方法

名称	描述
qspi_flash_init	Initialize QSPI-Flash Memory
change_mode_qspi_flash	Switch QSPI-Flash Memory mode between download and read, write, erase memory
qspi_flash_io_fast_read	Read data from QSPI-Flash Memory fastest, a single command and multi data
qspi_flash_fast_read	Read data from QSPI-Flash Memory fast, a single command and a single data
qspi_flash_write	Write data into QSPI-Flash Memory
qspi_flash_4ksector_erase	Erase 4KB sectors of QSPI-Flash Memory
qspi_flash_64ksector_erase	Erase 64KB sectors of QSPI-Flash Memory
qspi_flash_page_program	Write data into QSPI-Flash Memory with pages
qspi_flash_chip_erase	Erase full chip of QSPI-Flash Memory
qspi_flash_write_sr	Write status register of QSPI-Flash Memory
qspi_flash_read_sr	Read status register of QSPI-Flash Memory
qspi_flash_Enable	Enable QSPI-Flash Memory

~~17.2.2~~17.3.2 SPI-Flash 驱动程序

SPI-Flash Memory 驱动程序使用方法如表 17-3 所示。SPI-Flash Memory 驱动程序定义位于

library\libraries\drivers\src\GOWIN_M1_spi_flash.c。

表 17-3 SPI-Flash Memory 驱动程序使用方法

名称	描述
spi_flash_init	Initialize SPI-Flash Memory
change_mode_spi_flash	Switch SPI-Flash Memory mode between download and read, write, erase memory
spi_flash_read	Read data from SPI-Flash Memory
spi_flash_write	Write data into SPI-Flash Memory
spi_flash_4ksector_erase	Erase 4KB sectors of SPI-Flash Memory
spi_flash_page_program	Write data into SPI-Flash Memory with pages
spi_flash_64ksector_erase	Erase 64KB sectors of SPI-Flash Memory

18 PSRAM Memory

18.1 特征

Gowin_EMPU_M1，包含 1 个通过 AHB 总线访问的 PSRAM Memory 外设：

- AHB 总线接口
- 能与标准的 PSRAM Memory 器件接口兼容
- 支持存储器数据路径宽度为 8
- 支持 x8 数据宽度的内存芯片
- 可编程突发长度 32
- 时钟比例为 1:2
- 支持初始延时为 6
- 支持固定延时模式
- 支持电源关闭选项
- 驱动强度为 50
- 自刷新区域为 full
- 刷新速率为 normal

18.2 寄存器定义

PSRAM Memory 寄存器定义，如表 18-1 所示。PSRAM Memory 寄存器定义位于 library\libraries\cmsis\cm1\device_support\GOWIN_M1.h。

表 18-1 PSRAM Memory 寄存器定义

寄存器名称	地址偏移	类型	宽度	初始值	描述
CMD	0x00	RW	1	0x0	Command register [0] Operation type 0 = Read operation 1 = Write operation
ADDRESS	0x04	RW	21	0x0	Address register

寄存器名称	地址偏移	类型	宽度	初始值	描述
					[20:0] Address of reading and writing data
WR_DATA0	0x08	RW	32	0x0	Write data register 0 [31:0] Write first 32bit data
WR_DATA1	0x0C	RW	32	0x0	Write data register 1 [31:0] Write second 32bit data
WR_DATA2	0x10	RW	32	0x0	Write data register 2 [31:0] Write third 32bit data
WR_DATA3	0x14	RW	32		Write data register 3 [31:0] Write fourth 32bit data
CMD_EN	0x18	WO	1		Command enable register [0] Enable PSRAM
READ_DONE	0x1C	RW	1		Read status register [0] Read done flag, auto set 1 if it is done, and need mcu to clear
RD_DATA0	0x20	RO	32		Read data register 0 [31:0] Read first 32bit data
RD_DATA1	0x24	RO	32		Read data register 1 [31:0] Read second 32bit data
RD_DATA2	0x28	RO	32		Read data register 2 [31:0] Read third 32bit data
RD_DATA3	0x2C	RO	32		Read data register 3 [31:0] Read fourth 32bit data
INTI_DONE	0x30	RO	1		Initialization done register [0] PSRAM hardware initialization done flag 0 = Initialization failed 1 = Initialization done

18.3 驱动程序使用方法

PSRAM Memory 驱动程序使用方法，如表 18-2 所示。PSRAM Memory 驱动程序定义位于 library\libraries\drivers\src\GOWIN_M1_psram.c。

表 18-2 PSRAM Memory 驱动程序使用方法

名称	描述
PSRAM_Check_Init_Status	Check the status of PSRAM Memory initialization
PSRAM_Mode_Set	Set the mode fo PSRAM Memory write and read
PSRAM_Address_Set	Set the address of PSRAM Memory and save data into this address

名称	描述
PSRAM_Read_Data_Buff	Read data from the buffer of PSRAM Memory
PSRAM_Cmd_Enable	Enable the command of PSRAM Memory
PSRAM_Read_Done_Flag	Get the flag of read PSRAM Memory done
PSRAM_Clear_Read_Done_Flag	Clear the flag of read PSRAM Memory done
PSRAM_Write_Data_Buff	Write data into the buffer of PSRAM Memory
PSRAM_Cmd_Unable	Disable the command of PSRAM Memory
PSRAM_Write_Data_Package	Write a package data into PSRAM Memory
PSRAM_Read_Data_Package	Read a package data from PSRAM Memory

19 RTOS

Gowin_EMPU_M1 支持 uC/OS-III、FreeRTOS 和 RT-Thread Nano 版本 RTOS。

19.1 uC/OS-III

19.1.1 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；
- Gowin_EMPU_M1 支持 uC/OS-III；
- uC/OS-III 源代码请在 Micrium 网站 http://www.micrium.com 下载。

19.1.2 版本

Gowin_EMPU_M1 支持的 uC/OS-III 版本为 V3.03.00。

19.1.3 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。
- 用户可以通过修改 UCOS_BSP\bsp.c 和 bsp.h 来支持所用开发板。

19.2 FreeRTOS

19.2.1 特征

- FreeRTOS 是一个轻量级的实时操作系统；
- 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；
- Gowin_EMPU_M1 支持 FreeRTOS；
- FreeRTOS 源代码请在 FreeRTOS 网站 http://www.FreeRTOS.org 下载。

19.2.2 版本

Gowin_EMPU_M1 支持的 FreeRTOS 版本为 V10.2.1。

19.2.3 配置

用户可以通过修改 include\FreeRTOSConfig.h 来配置 FreeRTOS。

19.3 RT-Thread Nano 版本

19.3.1 特征

- RT-Thread Nano 是一个极简版的硬实时内核；
- 由 C 语言开发，采用面向对象的编程思维，具有良好的代码风格，是一款可裁剪的、抢占式实时多任务的 RTOS；
- 内存资源占用极小，功能包括任务处理、软件定时器、信号量、邮箱和实时调度等相对完整的实时操作系统特性；
- 开源免费，遵循 Apache 许可证 2.0，实时操作系统内核及所有开源组件可以免费在商业产品中使用，不需要公布应用程序源码，没有潜在商业风险。
- Gowin_EMPU_M1 支持 RT-Thread Nano；
- RT-Thread Nano 源代码请在 RT-Thread 网站 https://www.rt-thread.org 下载。

19.3.2 版本

Gowin_EMPU_M1 支持的 RT-Thread Nano 版本为 V3.1.5。

19.3.3 配置

- 用户可以通过修改 bsp\cm1\rtconfig.h 来配置 RT-Thread Nano。
- 用户可以通过修改 bsp\cm1\drivers\board.c 来支持所用开发板。

20 协议栈软件编程

Gowin_EMPU_M1 支持如下协议栈软件编程：TCP/IP 协议栈

20.1 TCP/IP 协议栈

Gowin_EMPU_M1 支持开源 TCP/IP 协议栈 LwIP。

LwIP 协议栈是一个轻量级的开源 TCP/IP 协议栈，专为嵌入式系统和小型设备而设计，旨在提供 TCP/IP 网络协议支持，使嵌入式系统能够与其他设备通过网络进行通信。

20.1.1 特征

- 轻量级：占用较少的内存和处理器资源，能够在嵌入式系统中高效运行
- 可裁剪：允许根据具体需求进行裁剪，只包含必要的协议和功能，以减少存储空间和处理器开销
- 高性能：使用性能优化技术，如零拷贝和事件驱动机制，以提高网络通信的效率和吞吐量
- 可移植性：具有良好的可移植性，可以在多种操作系统和硬件平台上运行，例如 RTOS FreeRTOS 和 uC/OS-III
- 多功能性：支持多种应用协议，除 TCP/IP 协议栈，还提供一些常用的应用层协议实现，如 HTTP、SNMP 和 MQTT
- Gowin_EMPU_M1 支持单片机、RTOS FreeRTOS 和 uC/OS-III 等模式的 LwIP 协议栈

20.1.2 版本

Gowin_EMPU_M1 支持的 LwIP 协议栈版本为 V2.1.2。

21 应用程序

Gowin_EMPU_M1 提供 ARM Keil MDK IDE（已测试软件版本：V5.26）和 GMD IDE（已测试软件版本：V1.2）软件环境的应用程序。

以下各节描述 Gowin_EMPU_M1 的各个应用程序。

21.1 UART

Gowin_EMPU_M1 提供 UART 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\printf

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\printf

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\uart_rx

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\uart_rx

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\uart_rx_intr

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\uart_rx_intr

21.2 Timer

Gowin_EMPU_M1 提供 Timer 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\timer

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\timer

21.3 Watch Dog

Gowin_EMPU_M1 提供 Watch Dog 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\watchdog

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\watchdog

21.4 GPIO

Gowin_EMPU_M1 提供 GPIO 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\led

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\led

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\gpio_in_intr

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\gpio_in_intr

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\keyscan

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\keyscan

21.5 I²C Master

Gowin_EMPU_M1 提供 I²C Master 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\i2c_master

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\i2c_master

21.6 SPI Master

Gowin_EMPU_M1 提供 SPI Master 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\spi_master

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\spi_master

21.7 RTC

Gowin_EMPU_M1 提供 RTC 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\rtc

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\rtc

21.8 TRNG

Gowin_EMPU_M1 提供 TRNG 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\tr

ng

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\trng

21.9 DualTimer

Gowin_EMPU_M1 提供 DualTimer 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\dualtimer

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\dualtimer

21.10 SD-Card

Gowin_EMPU_M1 提供 SD-Card 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_fatfs

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_fatfs

21.11 CAN

Gowin_EMPU_M1 提供 CAN 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\can

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\can

21.12 Ethernet

Gowin_EMPU_M1 提供 Ethernet 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\ethernet

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\ethernet

21.13 DDR3 Memory

Gowin_EMPU_M1 提供 DDR3 Memory 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\ddr3

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\ddr3

21.14 SPI-Flash Memory

Gowin_EMPU_M1 提供 SPI-Flash Memory 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\spi_flash

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\spi_flash

21.15 QSPI-Flash Memory

Gowin_EMPU_M1 提供 QSPI-Flash Memory 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\qspi_flash

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\qspi_flash

21.16 PSRAM Memory

Gowin_EMPU_M1 提供 PSRAM Memory 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\psram

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\psram

21.17 Interrupt

Gowin_EMPU_M1 提供 Interrupt 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\interrupt

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\interrupt

21.18 DMM

Gowin_EMPU_M1 提供 Dynamic Memory Management 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\dmm

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\dmm

21.19 AHB Master

Gowin_EMPU_M1 提供 AHB Master 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\ahb_master

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\ahb_master

21.20 APB Master

Gowin_EMPU_M1 提供 APB Master 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_demo\project\a

pb_master

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_demo\src\project\apb_master

21.21 uC/OS-III

Gowin_EMPU_M1 提供 RTOS uC/OS-III 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_ucos_iii

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_ucos_iii

21.22 FreeRTOS

Gowin_EMPU_M1 提供 RTOS FreeRTOS 应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_freertos

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_freertos

21.23 RT-Thread Nano 版本

Gowin_EMPU_M1 提供 RT-Thread Nano 版本应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_rtthread_nano

…\ref_design\MCU_RefDesign\GMD_RefDesign\cm1_rtthread_nano

21.24 LwIP 协议栈

Gowin_EMPU_M1 提供 LwIP 协议栈应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_tcpip\project\lwip_freertos_ping

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_tcpip\project\lwip_freertos_tcp_netcon_client

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_tcpip\project\lwip_freertos_tcp_socket_client

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_tcpip\project\lwip_freertos_tcpserver_socket

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_tcpip\project\lwip_raw_ping

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_tcpip\project\lwip_raw_udp

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_tcpip\project\lwip_uc3_client_ok

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_tcpip\project\lwip_uc3_ping_ok

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_tcpip\project\lwip_uc3_server_ok

21.25 uIP 协议栈

Gowin_EMPU_M1 提供 uIP 协议栈应用程序设计：

…\ref_design\MCU_RefDesign\MDK_RefDesign\cm1_tcpip\project\uip_helloworld


GOWIN高云
智慧逻辑 定制未来