

# WTM2101

Hal uart 使用说明

版本号: V1.0.0.0 日期: 2023.02.16



## 声明

#### 商标声明:



作为北京知存科技有限公司的商标,本文件中提到的所有其他

商标和商号均为其持有人的财产。

#### 版权声明:

Copyright © 2021 北京知存科技有限公司. All rights reserved.

#### 内容声明:

本文件中的信息如有更改, 恕不另行通知。为了确保内容的准确性, 文章会做出相关的确认, 但本文件中的所有声明、信息和建议不构成任何明示或暗示的保证。

#### 北京知存科技有限公司

地址:北京市海淀区北四环西路 56号辉煌时代大厦西座 1502

网址: <a href="http://www.witintech.com">http://www.witintech.com</a>

Page 2 of 12 WITMEM 保密文件



# 目录

<b>—</b> ,	文档功能说明	4
=,	函数功能介绍	5
	hal_uart_instance_get()	5
	hal_uart_init()	5
	hal_uart_open()	6
	hal_uart_ctl()	6
	hal_uart_write()	6
	hal_uart_read()	7
	hal_uart_close ()	7
三、	结构体介绍	
	Hal_Uart_InitTypeDef	8
	Hal Uart Gpio Typedef	9
	Hal_Uart_CacheTypeDef	9
	Hal_Uart_Dma_Typedef	9
四、	其他	. 11
Ŧ	修订历史	12



## 一、 文档功能说明

该文档旨在说明 hal\_uart 的驱动库.包括接口功能以及结构体参数意义.



Page 4 of 12 WITMEM 保密文件



#### 二、函数功能介绍

## hal\_uart\_instance\_get()

Hal\_Uart\_InitTypeDef\* hal\_uart\_instance\_get(Hal\_Uart\_Instance\_Typedef number) 获取 hal uart 实例对象

#### 参数

number: hal uart 实例对象编号

返回值

成功返回 Hal uart 实例对象,否则是 NULL

#### hal\_uart\_init()

int hal\_uart\_init(Hal\_Uart\_InitTypeDef \*uart\_instance,

UART\_TypeDef \*address,

UART\_InitTypeDef \*initstructure,

Hal\_Uart\_Receive\_Mode\_Typedef flag,

async handler async callback,

FunctionalState send\_dma\_enable\_flag)

初始化 hal uart 实例对象

#### 参数

uart instance: hal uart 实例对象

address: uart 硬件地址

initstructure: uart 参数配置结构体

flag: 同步或者异步接收标志位

async callback: 异步接收回调函数

send dma enable flag:发送 dma 使能标志.

#### 返回值

大于 0 成功,否则失败

#### 注意:

- 1. uart 硬件地址 UARTO 时,gpio 默认使用 gpio4, gpio5. uart 硬件地址 UART1 时,gpio 默认使用 gpio14, gpio15.
- 2. 使用发送 DMA 时,默认使用 DMA 通道 4
- 3. 接收 buffer 的大小默认 1536Byte

Page 5 of 12 WITMEM 保密文件



## hal\_uart\_open()

```
int hal uart open(Hal Uart InitTypeDef *uart instance)
根据传入的 hal uart 实例对象打开相应的硬件
参数
   uart instance:hal uart 实例对象
返回值
   大于 0 成功,否则失败
```

#### hal\_uart\_ctl()

```
int hal_uart_ctl(Hal_Uart_InitTypeDef *uart_instance,int command, ...)
控制 hal uart 实例对象或硬件参数
参数
   command:命令参数
   ...:命令参数的附加参数
返回值
   大于 0 成功,否则失败
注意:
```

命令参数: HAL\_UART\_MODE\_SWITCH\_COMMAND 表示切换串口接收异步或者同步

## hal\_uart\_write()

```
int hal_uart_write(Hal_Uart_InitTypeDef *uart_instance,
             uint8 t *buffer,
             uint32 t size);
通过 hal uart 实例对象的发送 buffer
参数
   uart instance:hal uart 实例对象
   buffer:数据 buffer
   size:数据 buffer 大小
返回值
   大于0成功,否则失败
```

WITMEM 保密文件 Page 6 of 12



### hal\_uart\_read()

## hal\_uart\_close ()

```
int hal_uart_close(struct Hal_Uart_InitTypeDef *uart_instance)

关闭 hal uart 实例对象及关联的硬件

参数

uart_instance:hal uart 实例对象

返回值

大于 0 成功,否则失败
```

Page 7 of 12 WITMEM 保密文件



#### 三、 结构体介绍

#### Hal\_Uart\_InitTypeDef

```
Hal uart 配置结构体
typedef struct Hal Uart InitTypeDef
   FunctionalState enable;
   UART TypeDef *instance;
   TMR TypeDef *timer instance;
   UART InitTypeDef initstructure;
   Hal Uart Gpio Typedefio;
   Hal Uart Receive Mode Typedef sync async flag;
   Bool sync receive flag;
    async handler async callback;
   Hal Uart Data handle data handle info;
   Hal Uart CacheTypeDef cache;
   Hal Uart Dma Typedef send dma;
}Hal Uart InitTypeDef;
参数
   enable:结构体初始化标志
   Instance:uart 硬件地址
   timer instance:timer 硬件地址,处理 uart 接收超时
   initstructure:uart 参数配置结构体
   io:uart 的 gpio 配置结构体
   sync async flag:uart 同步或异步设置标志位
   sync receive flag:uart 同步接收完成标志
   async callback:uart 异步回调函数接口
   data handle info:内部数据处理接口结构体
   cache:uart 接收 ring buffer 结构体
   send dma:uart 发送 dma 配置结构体
```

Page 8 of 12 WITMEM 保密文件



#### Hal\_Uart\_Gpio\_Typedef

```
Uart 配置结构体
typedef struct
{
    uint32_t uart_tx,uart_rx;
    uint32_t uart_tx_af,uart_rx_af;
}Hal_Uart_Gpio_Typedef;
参数
    uart_tx, uart_rx:uart 对应的 gpio
    uart_tx_af,uart_rx_af:uart 对应 gpio 的功能选择
```

## Hal\_Uart\_CacheTypeDef

```
Hal uart 缓存 buffer 结构体
typedef struct
{
    Ring_Cache ring_buffer;
}Hal_Uart_CacheTypeDef;
参数
    ring_buffer:环形缓存 buffer 结构体
```

## Hal\_Uart\_Dma\_Typedef

```
Hal uart 的 dma 配置结构体
typedef struct
{
    FunctionalState enable;
    uint32_t dma_channel;
    DMA_InitTypeDef config;
    DMA_LlpTypeDef *llp_cfg;
    volatile uint32_t finish_flag;
}Hal_Uart_Dma_Typedef;
参数
```

Page 9 of 12 WITMEM 保密文件



enable:dma 启动标志 dma\_channel:dma 使用的通道 config:dma 参数配置结构体 llp\_cfg:dma 链表模式对应的结构体



Page 10 of 12 WITMEM 保密文件



#### 四、 其他

1.uart 数据存储 buffer 申请自 heap.c 文件中的数组 static uint8\_t ucHeap[configTOTAL HEAP SIZE] attribute ((section(".audmem")));

在使用前需要自己在链接脚本中定义一块安全可用的地址,地址范围根据属性attribute ((section(".audmem")))来指定.数组大小由宏 configTOTAL HEAP SIZE 定义

2.在 hal\_uart.c 中的中断子程序只是示例代码,实际使用时务必重写中断子程序,否则会重现严重的错误.

Page 11 of 12 WITMEM 保密文件



## 五、 修订历史

表 5-1 修订历史

版本	日期	修订人	说明
V1.0.0.0	2023-02-16	李剑	初次编写

Page 12 of 12 WITMEM 保密文件