



RTOS UART 开发指南

版本号: 1.1
发布日期: 2021.4.14

版本历史

版本号	日期	制/修订人	内容描述
1.0	2020.7.7	AWA1636	1. 初版
1.1	2021.4.14	AWA1636	1. 增加 sys_config.fex



目 录

1 前言	1
1.1 文档简介	1
1.2 目标读者	1
1.3 适用范围	1
2 模块介绍	2
2.1 模块功能介绍	2
2.2 相关术语介绍	2
2.2.1 硬件术语	2
2.2.2 软件术语	2
2.3 模块配置介绍	2
2.3.1 platform 配置说明	2
2.3.2 pin 引脚配置	3
2.3.3 kernel menuconfig 配置说明	3
2.4 源码结构介绍	4
2.5 驱动框架介绍	4
3 模块接口说明	5
3.1 hal_uart_get_version	5
3.2 hal_uart_init	6
3.3 hal_uart_deinit	6
3.4 hal_uart_send	6
3.5 hal_uart_receive	6
3.6 hal_uart_receive_polling	7
3.7 hal_uart_set_hardware_flowcontrol	7
3.8 hal_uart_disable_flowcontrol	7
4 模块使用范例	9
5 FAQ	10

1 前言

1.1 文档简介

介绍 RTOS 中 UART 驱动的接口及使用方法，为 UART 设备的使用者提供参考。

1.2 目标读者

UART 驱动、及应用层的开发/维护人员。

1.3 适用范围

表 1-1: 适用产品列表

产品名称	内核版本	驱动文件
F133	Melis	hal_uart.c
V833	Melis	hal_uart.c
R328	FreeRTOS	hal_uart.c
R329-DSP	FreeRTOS	hal_uart.c

2 模块介绍

2.1 模块功能介绍

BSP UART 驱动主要实现设备驱动的底层细节，并为上层提供一套标准的 API 接口以供使用。

2.2 相关术语介绍

2.2.1 硬件术语

术语	解释说明
UART	Universal Asynchronous Receiver/Transmitter, 通用异步收发传输器

2.2.2 软件术语

术语	解释说明
HAL	Hardware Abstraction Layer, 硬件抽象层
RTOS	Real Time Operating System, 实时操作系统
GPIO	General Purpose Input/Output, 通用输入输出

2.3 模块配置介绍

2.3.1 platform 配置说明

在不同的 Sunxi 硬件平台中，UART 控制器的数目也不同，每个 UART 控制器支持的线数也不同，但平台配置文件的信息基本类似，如下：

```

#define SUNXI_IRQ_UART0          (81) /* 108 uart0 interrupt */
#define SUNXI_IRQ_UART1          (82) /* 109 uart1 interrupt */
#define SUNXI_IRQ_UART2          (83) /* 110 uart2 interrupt */
#define SUNXI_IRQ_UART3          (84) /* 111 uart3 interrupt */

//base register infomation
#define SUNXI_UART0_BASE         (0x05000000)
#define SUNXI_UART1_BASE         (0x05000400)
#define SUNXI_UART2_BASE         (0x05000800)
#define SUNXI_UART3_BASE         (0x05000c00)

#define UART_FIFO_SIZE           (256)
#define UART_GPIO_FUNCTION        (5)

```

说明

1. **SUNXI_IRQ_UART0**, 表示 **UART** 中断号;
2. **SUNXI_UART0_BASE**, 表示 **UART** 寄存器基地址;
3. **UART_FIFO_SIZE**, **UART** 的 **FIFO** 大小;
4. **UART_GPIO_FUNCTION**, **UART** 的引脚复用功能;

2.3.2 pin 引脚配置

引脚配置在 source/project/方案/configs/sys_config_xxx.fex 中存放 (xx 是存储方案的标识, 例如 sys_config_nor.cfg、sys_config_nand.cfg)。

```

[uart0]
uart_tx      = port:PB08<6><1><default><default>
uart_rx      = port:PB09<6><1><default><default>

```

说明

引脚说明: **port**: 端口 <复用功能><上下拉><驱动能力><输出值>

2.3.3 kernel menuconfig 配置说明

路径如下:

```

Kernel Setup --->
  Drivers Setup --->
    SoC HAL Drivers --->
      UART Devices --->

```

```

[*] enable uart driver
[*]   enable uart hal APIs test command

```

图 2-1: UART menuconfig

2.4 源码结构介绍

```
hal/source/uart/ ---- 驱动源码
├── hal_uart.c
├── Kconfig
├── Makefile
├── platform
│   ├── uart-sun8iw18.h
│   ├── uart-sun8iw19.h
│   ├── uart-sun8iw20.h
│   └── ...
└── platform-uart.h
└── uart.h

include/hal/ ---- 驱动APIs声明头文件
└── hal_uart.h

hal/test/uart/ ---- 驱动APIs测试代码
├── Makefile
└── test_uart.c
```

2.5 驱动框架介绍

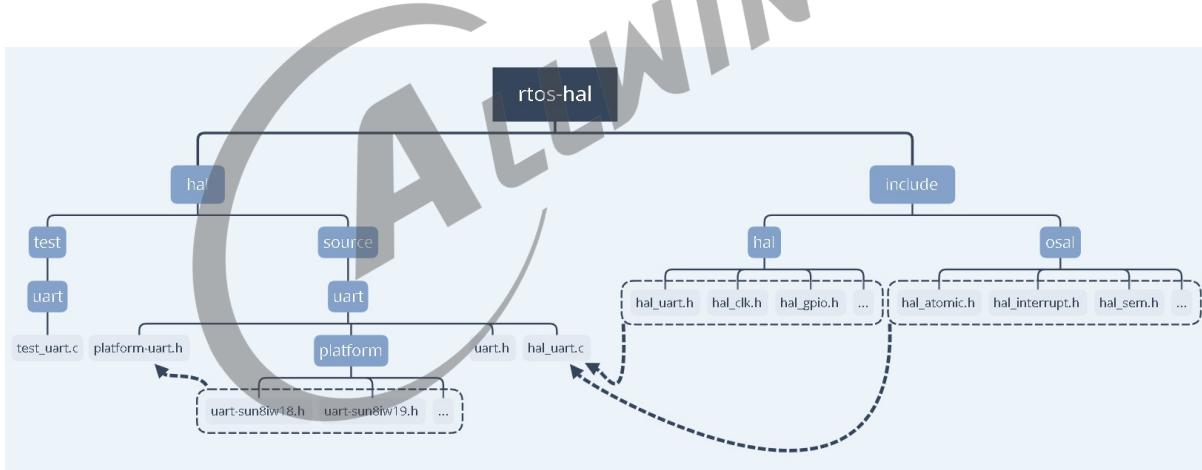


图 2-2: UART 驱动框架

3 模块接口说明

需包含头文件：

```
#include <hal_uart.h>
```

API	解释说明
hal_uart_get_version	获取 uart 驱动版本号
hal_uart_get_capabilities	获取 uart 驱动包含的 features (预留)
hal_uart_init	初始化 uart 驱动
hal_uart_deinit	卸载 uart 驱动
hal_uart_power_control	(预留)
hal_uart_send	发送处理
hal_uart_receive	接收处理
hal_uart_get_tx_count	(预留)
hal_uart_get_rx_count	(预留)
hal_uart_get_status	(预留)
hal_uart_set_modem_control	(预留)
hal_uart_get_modem_status	(预留)
hal_uart_receive_polling	轮训接收
hal_uart_set_hardware_flowcontrol	设置硬件流控
hal_uart_disable_flowcontrol	禁止硬件流控

3.1 hal_uart_get_version

- 原型：sunxi_hal_version_t hal_uart_get_version(int32_t dev)
- 作用：获取 UART 驱动的版本号
- 参数：
 - dev:UART 端口号
- 返回：
 - 非空:UART 驱动版本号

3.2 hal_uart_init

- 原型: `int32_t hal_uart_init(int32_t uart_port)`
- 作用: 初始化 UART 驱动
- 参数:
 - `uart_port`: UART 端口号
- 返回:
 - `SUNXI_HAL_OK`: 成功
 - `HAL_UART_STATUS_ERROR`: 失败

3.3 hal_uart_deinit

- 原型: `int32_t hal_uart_deinit(int32_t uart_port)`
- 作用: 卸载 UART 驱动
- 参数:
 - `uart_port`: UART 端口号
- 返回:
 - `SUNXI_HAL_OK`: 成功

3.4 hal_uart_send

- 原型: `int32_t hal_uart_send(int32_t dev, const uint8_t *data, uint32_t num)`
- 作用: 发送处理
- 参数:
 - `dev`: UART 端口号
 - `data`: 发送数据缓冲区
 - `num`: 发送数据长度
- 返回:
 - `size`: 成功发送的字节数

3.5 hal_uart_receive

- 原型: `int32_t hal_uart_receive(int32_t dev, uint8_t *data, uint32_t num)`

- 作用：接收处理
- 参数：
 - dev:UART 端口号
 - data: 接收数据缓冲区
 - num: 接收数据长度
- 返回：
 - size: 成功接收的字节数

3.6 hal_uart_receive_polling

- 原型：int32_t hal_uart_receive_polling(int32_t dev, uint8_t *data, uint32_t num)
- 作用：轮询接收
- 参数：
 - dev:UART 端口号
 - data: 接收数据缓冲区
 - num: 接收数据长度
- 返回：
 - size: 成功接收的字节数

3.7 hal_uart_set_hardware_flowcontrol

- 原型：void hal_uart_set_hardware_flowcontrol(uart_port_t uart_port)
- 作用：设置硬件流控
- 参数：
 - uart_port:UART 端口号
- 返回：
 - void

3.8 hal_uart_disable_flowcontrol

- 原型：void hal_uart_disable_flowcontrol(uart_port_t uart_port)
- 作用：禁止硬件流控
- 参数：
 - uart_port:UART 端口号

- 返回：

- void



4 模块使用范例

可参考驱动 APIs 测试代码 (hal/test/uart/) 。



5 FAQ

无。



著作权声明

版权所有 © 2021 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明

  **全志科技**  (不完全列举) 均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。