

Sunxi UART 驱动 使用文档

珠海全志科技股份有限公司chenmingkun

珠海全志科技股份有限公司chenmingkun

珠海全志科技股份有限公司chenmingkun

文档履历

版本号	日期	制/修订人	内容描述
V1.0	2020-03-11	班涛	建立初版

珠海全志科技股份有限公司chenmingkun

珠海全志科技股份有限公司chenmingkun

珠海全志科技股份有限公司chenmingkun

目 录

1. 概述.....	1
1.1. 编写目的.....	1
1.2. 适用范围.....	1
1.3. 相关人员.....	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
2.3. 模块配置介绍.....	2
2.3.1 配置说明.....	2
2.4. 源码结构介绍.....	3
3. 接口描述.....	4
3.1. 应用层接口.....	4
3.1.1. 打开/关闭串口.....	4
3.1.2. 读/写串口.....	4
3.1.3. 设置串口属性.....	4
3.2. 驱动层接口.....	5
3.2.1. 打开/关闭串口.....	5
3.2.2. 读/写串口.....	5
3.2.3. 设置串口属性.....	6
4. DEMO.....	7

珠海全志科技股份有限公司chenmingkun

珠海全志科技股份有限公司chenmingkun

珠海全志科技股份有限公司chenmingkun

1. 概述

1.1. 编写目的

介绍 Melis-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. 适用范围

适用于 Allwinnertech 的 AW1817 系列平台。

1.3. 相关人员

UART 驱动、及应用层的开发/维护人员。

珠海全志科技股份有限公司chenmingkun

珠海全志科技股份有限公司chenmingkun

珠海全志科技股份有限公司chenmingkun

2. 模块介绍

2.1. 模块功能介绍

Melis 中，UART 驱动的结构可以分为两个层次：

1. Hal_uart，为 UART 驱动提供了一套 API，实现设备驱动底层细节；
2. Drv_uart，实现了 RTOS 中所有 UART 设备的注册和管理。

2.2. 相关术语介绍

术语	解释说明
Sunxi	指 Allwinner 的一系列 SOC 硬件平台。
UART	Universal Asynchronous Receiver/Transmitter，通用异步收发传输器。

2.3. 模块配置介绍

2.3.1 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。线数不同在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 APB_USART0_BASE      (0x05000000)
#define APB_USART1_BASE      (0x05000400)
#define APB_USART2_BASE      (0x05000800)
#define APB_USART3_BASE      (0x05000c00)

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

#define UART0_TX      GPIOH(9)
#define UART0_RX      GPIOH(10)

#define UART1_TX      GPIOG(6)
#define UART1_RX      GPIOG(7)

#define UART2_TX      GPIOH(5)
#define UART2_RX      GPIOH(6)

#define UART3_TX      GPIOH(0)
```

```
#define UART3_RX GPIOH(1)
```

其中：

1. SUNXI_IRQ_UART0, 表示 UART 中断号;
2. APB_USART0_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;

2.4. 源码结构介绍

UART 驱动的源代码位于内核在 `drivers/tty/serial` 目录下：

`source/ekernel/drivers/hal/source/serial/`

└── `hal_serial.c` // Sunxi 平台的 UART 控制器驱动代码

`source/ekernel/drivers/include/hal/`

└── `sunxi_hal_uart.h` // 为 Sunxi 平台的 UART 控制器驱动定义了一些宏、数据结构

`source/ekernel/drivers/drv/source/serial/`

└── `uart_drv.c` // 驱动层源码

`source/ekernel/drivers/test/`

└── `test_uart.c` // 为 Sunxi 平台的 UART 控制器测试文件

珠海全志科技股份有限公司chenmingkun

珠海全志科技股份有限公司chenmingkun

珠海全志科技股份有限公司chenmingkun

3. 接口描述

UART 应用层的使用类似 Linux 系统中的标准串口编程方法。

3.1. 应用层接口

3.1.1. 打开/关闭串口

使用标准的文件打开函数：

```
fd = rt_device_find("uart3");  
rt_device_open(fd, RT_DEVICE_OFLAG_RDWR | RT_DEVICE_FLAG_STREAM);  
rt_device_close((rt_device_t)fd);
```

需要引用头文件：

```
#include <sunxi_drv_uart.h>
```

3.1.2. 读/写串口

同样使用标准的文件读写函数：

```
rt_device_write((rt_device_t)fd, 0, tbuf, len);  
rt_device_read((rt_device_t)fd, 0, rbuf, len);
```

需要引用头文件：

```
#include <sunxi_drv_uart.h>
```

3.1.3. 设置串口属性

串口属性包括波特率、数据位、停止位、校验位等，这部分是串口设备特有的接口。串口属性的数据结构定义如下：（sunxi_hal_usart.h）

```
typedef struct  
{  
    uart_baudrate_t baudrate;  
    uart_word_length_t word_length;  
    uart_stop_bit_t stop_bit;  
    uart_parity_t parity;  
}_uart_config_t;
```

【baudrate】：波特率，取值是一组宏，定义在 sunxi_hal_usart.h:

```
typedef enum  
{  
    UART_BAUDRATE_300 = 0,  
    UART_BAUDRATE_600,
```

```
UART_BAUDRATE_1200,  
UART_BAUDRATE_2400,  
UART_BAUDRATE_4800,  
UART_BAUDRATE_9600,  
UART_BAUDRATE_19200,  
UART_BAUDRATE_38400,  
UART_BAUDRATE_57600,  
UART_BAUDRATE_115200,  
UART_BAUDRATE_230400,  
UART_BAUDRATE_576000,  
UART_BAUDRATE_921600,  
UART_BAUDRATE_1000000,  
UART_BAUDRATE_1500000,  
UART_BAUDRATE_3000000,  
UART_BAUDRATE_4000000,  
UART_BAUDRATE_MAX,  
} uart_baudrate_t;
```

3.1.3.1. rt_device_control()

【函数原型】: `rt_err_t rt_device_control(rt_device_t dev, int cmd, void *arg)`

【功能描述】: 设置串口设备的属性。

【参数说明】: `fd`, 串口设备的文件描述符

`cmd`, 暂无使用

`arg`, 用于保存串口属性, 为 `_uart_config_t` 结构体

【返回值】: 0, 成功; -1, 失败, `errno` 给出具体错误码

3.2. 驱动层接口

3.2.1. 打开/关闭串口

```
hal_uart_initialize(port);
```

```
hal_uart_uninitialize(port);
```

需要引用头文件:

```
#include <sunxi_hal_usart.h>
```

3.2.2. 读/写串口

```
hal_uart_send(port, tbuf, len);
```



```
hal_uart_receive(port, rbuf, len);
```

需要引用头文件:

```
#include <sunxi_hal_usart.h>
```

3.2.3. 设置串口属性

```
hal_uart_control(port, NULL, &uart_config);
```

【函数原型】: int32_t hal_uart_control(int32_t uart_port, int cmd, void *args)

【功能描述】: 设置串口设备的属性。

【参数说明】: fd, 串口设备的文件描述符

cmd, 暂无使用

args, 用于保存串口属性, 为_uart_config_t 结构体

【返回值】: 0, 成功; -1, 失败, errno 给出具体错误码

珠海全志科技股份有限公司chenmingkun

珠海全志科技股份有限公司chenmingkun

珠海全志科技股份有限公司chenmingkun

4. demo

Demo 程序在 source/ekernel/drivers/test/test_uart.c

此 demo 程序是打开一个串口设备。

使用方法一：

物理短接 uart3 的 tx 和 rx 引脚。

终端输入：drvuart

Uart3 发送 “hello” 字符，波特率默认 9600。

使用方法二：

终端输入：uart 3 115200 -w

Uart3 发送 “hello” 字符，波特率设置为 115200。

珠海全志科技股份有限公司chenmingkun

珠海全志科技股份有限公司chenmingkun

珠海全志科技股份有限公司chenmingkun