



**APB-UART**

2024.08.27

# 一 修订

版本	日期	编辑人	内容
1.00	2024.08.27	陈家耀	创建了第一个正式版本
1.01	2025.03.06	陈家耀	修改 UART 中断方式接收例程

## 二 简介和特性

APB-UART 是一个带有 APB 从接口的 UART 控制器，可直接挂载在 APB 总线上充当 UART 外设。本 IP 简单易用、资源消耗少，具有以下特性：

- 可配置的 UART 波特率
- 使用收发 FIFO 来缓存待发送的字节数据和接收到的字节数据
- 支持 UART 收发中断

APB-UART 的组成如图 2-1 所示，它由 **APB 寄存器接口**、**中断控制**、**收发 FIFO** 和 **UART 收发控制** 四部分组成。**UART 发送控制** 从发送 FIFO 载入待发送的字节数据，根据波特率进行分频，产生起始位、数据位和停止位。**UART 接收控制** 首先检测起始位，然后根据波特率延迟半个 UART 位以对齐到起始位中央，接着每隔一个 UART 位移入一个数据位，最后等待一个 UART 位以对齐到停止位中央。

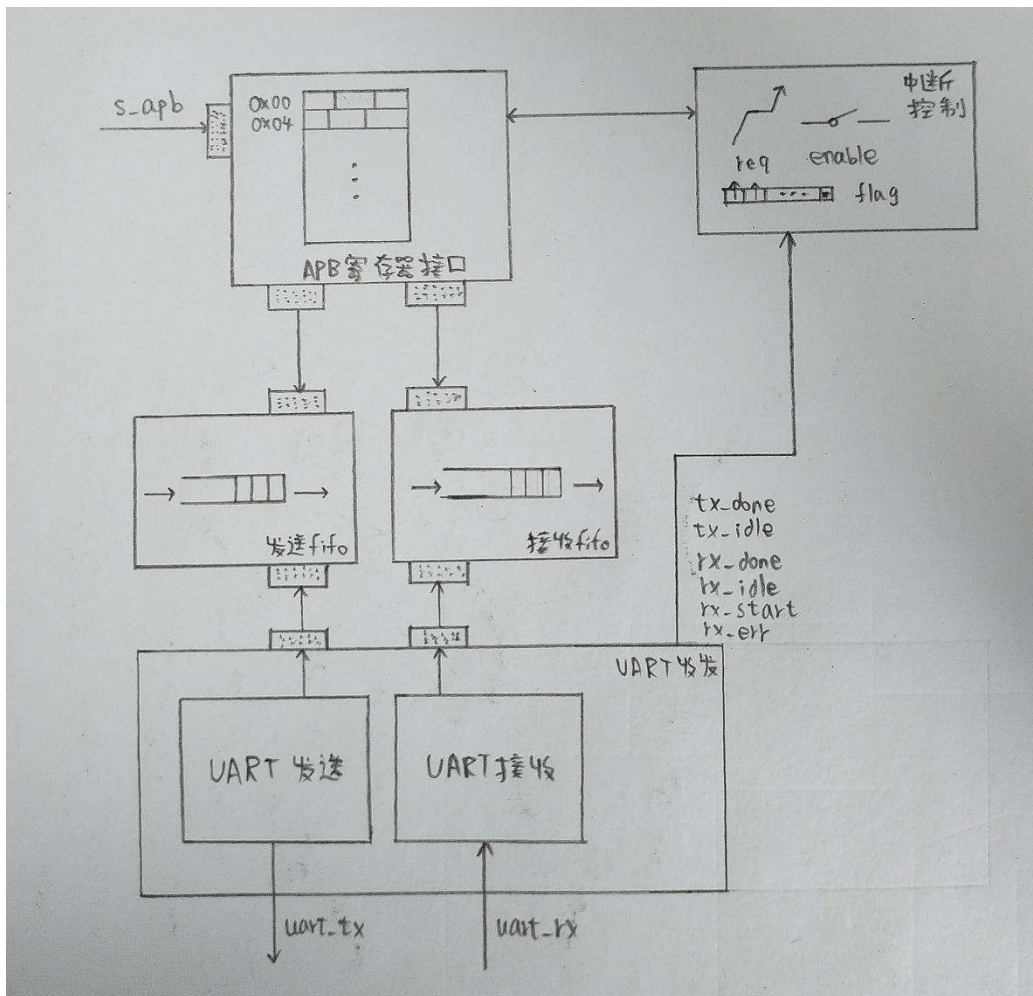


图 2-1 APB-UART 组成框图

### 三 IP 功能

APB-UART 是通用的 UART 外设，能够控制 UART 收发，可启用 UART 收发中断。其功能描述如下：

1、**通过 UART 发送字节数据**。用户通过 APB 寄存器接口将待发送的字节数据写入发送 FIFO，UART 发送控制模块自动从发送 FIFO 取数据，并通过 UART 接口发送该数据。

2、**通过 UART 接收字节数据**。UART 接收控制模块自动从 UART 接口接收数据，并存入接收 FIFO。用户可通过 APB 寄存器接口从接收 FIFO 获取数据。

3、**可启用的 UART 收发中断**。APB-UART 支持 2 种发送中断（**UART 发送达到规定字节数中断**、**UART 发送 IDLE 中断**）和 3 种接收中断（**UART 接收达到规定字节数中断**、**UART 接收 IDLE 中断**、**UART 接收 FIFO 溢出中断**）。**UART 收发中断字节数阈值**和 **IDLE 周期数阈值**均可通过 APB 寄存器接口进行配置。

4、**可配置的 UART 波特率**。APB-UART 根据用户提供的时钟频率和波特率来进行分频，以满足 UART 时序要求。

## 四 IO 描述

表 4-1 APB-UART IO 表

端口名	方向	位宽	含义
时钟和复位			
clk	input	1	时钟
resetrn	input	1	复位，低有效
APB 从接口			
paddr	input	32	APB 从机地址
psel	input	1	APB 从机片选
penable	input	1	APB 从机传输使能
pwrite	input	1	APB 从机读写类型
pwrdata	input	32	APB 从机写数据
pready_out	output	1	APB 从机传输完成，固定为 1
prdata_out	output	32	APB 从机读数据
pslverr_out	output	1	APB 从机传输错误，固定为 0
UART 接口			
uart_tx	output	1	UART 发送
uart_rx	input	1	UART 接收，应当连接上拉电阻
中断信号			
uart_itr	output	1	UART 外设中断请求

## 五 可配置参数描述

表 5-1 APB-GPIO 可配置参数表

配置参数名	含义	可取值
clk_frequency_MHz	时钟频率	32 位无符号整型，以 MHz 计
baud_rate	波特率	32 位无符号整型
tx_rx_fifo_ram_type	收发的 RAM 类型	"lutram"   "bram"
tx_fifo_depth	发送 fifo 深度	32   64   128   ...   2048
rx_fifo_depth	接收 fifo 深度	32   64   128   ...   2048
en_itr	是否使能 UART 收发中断	"true"   "false"
simulation_delay	仿真延时，可用于仿真时模拟 D 到 Q 延迟	0.1f~100.0f

## 六 应用指南

### 6.1 RTL 设计指南

APB-UART 是标准的 APB 外设，请将 APB-UART 挂载在 APB 总线上使用，典型情况是挂载在 AXI-APB 桥或 AHB-APB 桥上作为一个 APB 从机，如图 6-1-1 所示。关于 AXI-APB 桥或 AHB-APB 桥，请参见 UG200。

本 IP 所提供的同步 FIFO 的顶层 RTL 文件为 `ram_fifo_wrapper.v`，由于 fifo 使用到的 RAM 可能与器件类型有关，必要时请根据设计要求进行替换。

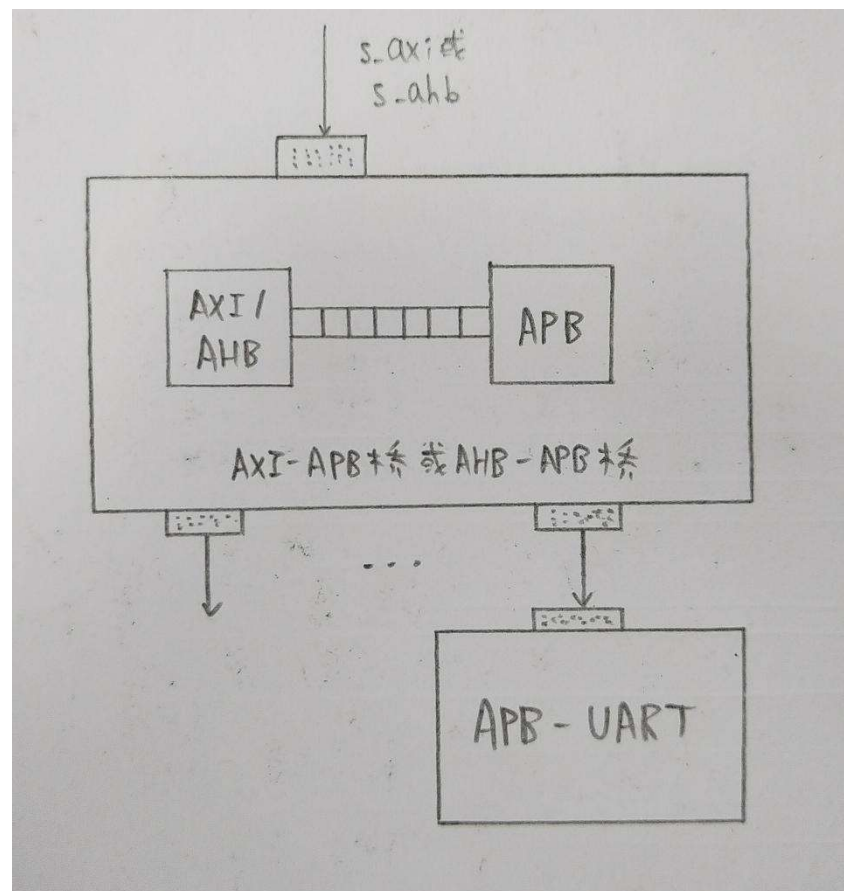


图 6-1-1 APB-UART 应用图

### 6.2 软件编程指南

#### 6.2.1 软件驱动 API

##### 1、类型定义

- ApbUART 结构体（APB-UART 外设结构体）

**ApbUARTHd\* hardware:** APB-UART 寄存器接口结构体指针，映射到 UART 外设的寄存器接口

**uint8\_t itr\_en:** 当前的中断使能向量

- ApbUARTDd 结构体（APB-UART 寄存器接口结构体）
  - uint32\_t fifo\_cs:** 收发 fifo 控制
  - uint32\_t itr\_status\_en:** 中断控制
  - uint32\_t tx\_itr\_th:** 发送中断阈值
  - uint32\_t rx\_itr\_th:** 接收中断阈值
- ApbUartItrThConfig 结构体（APB-UART 中断阈值配置结构体）
  - uint16\_t tx\_bytes\_n\_th:** UART 发送中断字节数阈值
  - uint16\_t tx\_idle\_th:** UART 发送中断 IDLE 周期数阈值
  - uint16\_t rx\_bytes\_n\_th:** UART 接收中断字节数阈值
  - uint16\_t rx\_idle\_th:** UART 接收中断 IDLE 周期数阈值

## 2、宏定义

- 中断类型掩码
  - APB\_UART\_TX\_BYTES\_N\_ITR\_MASK:** UART 发送达到规定字节数中断
  - APB\_UART\_TX\_IDLE\_ITR\_MASK:** UART 发送 IDLE 中断
  - APB\_UART\_RX\_BYTES\_N\_ITR\_MASK:** UART 接收达到规定字节数中断
  - APB\_UART\_RX\_IDLE\_ITR\_MASK:** UART 接收 IDLE 中断
  - APB\_UART\_RX\_ERR\_ITR\_MASK:** UART 接收 FIFO 溢出中断

## 3、函数

- void apb\_uart\_init(ApbUART\* uart, uint32\_t base\_addr);
  - 简介：初始化 APB-UART
  - 参数：**uart** APB-UART（结构体指针）
  - base\_addr** APB-UART 外设基地址
  - 返回值：无
- int apb\_uart\_send\_byte(ApbUART\* uart, uint8\_t byte);
  - 简介：APB-UART 发送一个字节
  - 参数：**uart** APB-UART（结构体指针）
  - byte** 待发送的字节数据
  - 返回值：是否成功
- int apb\_uart\_rev\_byte(ApbUART\* uart, uint8\_t\* byte);
  - 简介：APB-UART 获取一个接收字节
  - 参数：**uart** APB-UART（结构体指针）
  - byte** 接收字节数据缓冲区（首地址）
  - 返回值：是否成功
- void apb\_uart\_enable\_itr(ApbUART\* uart, uint8\_t itr\_mask, const ApbUartItrThConfig\* config);
  - 简介：APB-UART 使能中断
  - 参数：**uart** APB-UART（结构体指针）
  - itr\_mask** 中断使能向量
  - config** 收发中断阈值配置（结构体指针）



返回值：无

• void apb\_uart\_disable\_itr(ApbUART\* uart);

简介：APB-UART 除能中断

参数：uart APB-UART（结构体指针）

返回值：无

• uint8\_t apb\_uart\_get\_itr\_status(ApbUART\* uart);

简介：APB-UART 获取中断状态

参数：uart APB-UART（结构体指针）

返回值：中断状态

• void apb\_uart\_clear\_itr\_flag(ApbUART\* uart);

简介：APB-UART 清除中断标志

参数：uart APB-UART（结构体指针）

返回值：无

• void uart\_printf(ApbUART\* uart, char \*fmt, ...);

简介：APB-UART 格式化发送字符串

参数：uart APB-UART（结构体指针）

fmt 格式化字符串

... 字符串附加参数

返回值：无

## 6.2.2 软件编程示例

### 1、UART 发送

本示例基于 APB-UART 发送了字符串"hello, world!\n\r"。

```
1.  /**
   ****
2.  APB-UART 示例代码
3.  @brief UART 发送示例
4.  @date 2024/08/28
5.  @author 陈家耀
6.  @eidt 2024/08/28 1.00 创建了第一个正式版本
7.  ****
   ****/
8.
9.  #include "../apb_uart.h"
10.
11.  //////////////////////////////////////
12.
13.  #define APB_UART_BASEADDR 0x40000000 // APB-UART 外设基地址
14.
```

```

15.  //////////////////////////////////////
16.
17.  static ApbUART uart; // APB-UART 外设结构体
18.
19.  //////////////////////////////////////
20.
21.  void apb_uart_tx_example(void){
22.      apb_uart_init(&uart, APB_UART_BASEADDR); // 初始化 APB-UART
23.
24.      // 发送字符串
25.      uart_printf(&uart, "hello, world!\n\r");
26.
27.      while(1);
28.  }

```

## 2、UART 接收中断

本示例启用了 UART 接收 IDLE 中断，在中断服务函数中收集每个 UART 数据包并按原样发回。

```

1.  /*****
   *****/
2.  APB-UART 示例代码
3.  @brief UART 接收 IDLE 中断示例
4.  @attention 请根据硬件平台更换与全局中断控制器相关的 API
5.  @date 2024/08/28
6.  @author 陈家耀
7.  @eidt 2024/08/28 1.00 创建了第一个正式版本
8.  *****/
9.
10. #include "../apb_uart.h"
11.
12. #include "CMSDK_CM0.h"
13.
14. //////////////////////////////////////
15.
16. #define APB_UART_BASEADDR 0x40000000 // APB-UART 外设基地址
17.
18. //////////////////////////////////////
19.
20. static ApbUART uart; // APB-UART 外设结构体
21.
22. //////////////////////////////////////
23.

```

```

24.  /*****
25.  @itr_handler
26.  @private
27.  @brief APB-UART 收发中断服务程序(示例)
28.  @param none
29.  @return none
30.  *****/
31.  void USER_UART_Handler(void){
32.      uint8_t byte;
33.
34.      while(!apb_uart_rev_byte(&uart, &byte)){ // 收集 UART 数据包
35.          apb_uart_send_byte(&uart, byte); // 按原样发回
36.      }
37.
38.      apb_uart_clear_itr_flag(&uart); // 清零中断标志
39.  }
40.
41.  //////////////////////////////////////
42.
43.  void apb_uart_rx_itr_example(void){
44.      apb_uart_init(&uart, APB_UART_BASEADDR); // 初始化 APB-UART
45.
46.      // 配置 UART 接收 IDLE 中断
47.      const ApbUartItrThConfig uart_config = {10, 100, 10, 100}; // UART 接收中断 IDLE 周期数阈值 = 100
48.
49.      apb_uart_enable_itr(&uart, APB_UART_RX_IDLE_ITR_MASK, &uart_config); // 使能 UART 接收 IDLE 中断
50.
51.      NVIC_SetPriority((IRQn_Type)4, 0x03); // NVIC 设置 4 号中断优先级
52.      NVIC_EnableIRQ((IRQn_Type)4); // NVIC 使能 4 号中断
53.
54.      while(1);
55.  }

```