



UART 应用说明

V1.0

Yichip Microelectronics

©2020

Revision History

Version	Date	Author	Description
1.0	2020-12-04	qin.su	Initial version

目录

1 文档说明	4
1.1 编写目的	4
1.2 适用范围	4
1.3 文件说明	4
2 结构体及枚举说明	4
2.1 串口初始化结构体(USART_InitStruct)	4
3 函数说明	6
3.1 USART_Init	6
3.2 USART_DeInit	7
3.3 USART_SendData	7
3.4 USART_ReceiveData	7
3.5 USART_GetRxCount	8
3.6 USART_ReadDatatoBuff	8
3.7 USART_SendDataFromBuff	8
3.8 USART_SetRxITNum	错误!未定义书签。
3.9 USART_SetRxTimeout	错误!未定义书签。
4 Demo 函数说明	9
4.1 UARTx_Configuration	错误!未定义书签。
4.2 USART_SendDataFromBuff	错误!未定义书签。

1 文档说明

1.1 编写目的

为使用 UART 相关 Demo 及相关 UART 库函数提供说明。

1.2 适用范围

YC11xx 系列芯片。

1.3 文件说明

UART Demo 路径为: `cm0\ModuleDemo\uart`
UART 库文件为如下图 `yc11xx_uart.c` 与 `yc11xx_uart.h`,路径为 :
`cm0\Librarier\drivers\uart`

本 demo 主要介绍了串口的收发。

2 宏、结构体及枚举说明

2.1 宏

A. UART 波特率定义说明

- 0-13 bit 表示 `uart` 波特率分频系数，波特率=`uart` 时钟/分频系数。
- 第 15bit 表示 `uart dma buff` 地址来源。1: `m0` 地址。0: `bt` 核地址

```
#define UARTE_BAUDRATE_BAUDRATE_Baud4800      (0xa710)
#define UARTE_BAUDRATE_BAUDRATE_Baud9600      (0x9388)
#define UARTE_BAUDRATE_BAUDRATE_Baud14400     (0x8d05)
#define UARTE_BAUDRATE_BAUDRATE_Baud19200     (0x89c4)
#define UARTE_BAUDRATE_BAUDRATE_Baud38400     (0x84e2)
#define UARTE_BAUDRATE_BAUDRATE_Baud56000     (0x8359)
#define UARTE_BAUDRATE_BAUDRATE_Baud57600     (0x8341)
#define UARTE_BAUDRATE_BAUDRATE_Baud76800     (0x8271)
#define UARTE_BAUDRATE_BAUDRATE_Baud115200    (0x81A0)
#define UARTE_BAUDRATE_BAUDRATE_Baud230400    (0x80d0)
#define UARTE_BAUDRATE_BAUDRATE_Baud256000    (0x80bb)
```

```
#define UARTE_BAUDRATE_BAUDRATE_Baud460800    (0x8068)
#define UARTE_BAUDRATE_BAUDRATE_Baud921600    (0x8034)
#define UARTE_BAUDRATE_BAUDRATE_Baud1M        (0x9000)
```

B. UART 数据宽度定义

```
#define USART_WordLength_8b                    (0<<2)
#define USART_WordLength_9b                    (1<<2)
```

C. UART 停止位定义

```
#define USART_StopBits_1                      (0<<3)
#define USART_StopBits_2                      (1<<3)
```

D. UART 奇偶校验位定义

```
#define USART_Parity_Even                     (0<<1)
#define USART_Parity_Odd                     (1 << 1)
```

E. UART 模式定义

```
#define USART_Mode_Single_Line                (1<<6)
#define USART_Mode_duplex                     (0<<6)
```

F. UART 流控定义

```
#define USART_HardwareFlowControl_None        (0<<4)
#define USART_HardwareFlowControl_ENABLE     (1<<4)
```

G. DMA BUF 接受长度定义

```
#define uart_DMA_buf_len 1024
```

2.2 uart 枚举说明

枚举名称	说明
UARTA	串口 1
UARTB	串口 2

2.3 串口初始化结构体(USART_InitTypeDef)

元素名称	数据类型	说明	参数项
USART_BaudRate	Uin32_t	波特率	默认 115200
USART_WordLength	uint16_t	数据宽度	Databits_8b (8bit, 则无奇偶校验位) Databits_9b (9bit, 第 9bit 为奇偶校验位)
USART_StopBits	uint16_t	停止位	StopBits_1 (0) StopBits_2 (1)
USART_Parity	uint16_t	奇偶校验位	Parity_Even (偶校验) Parity_Odd (奇校验)
USART_Mode	uint16_t	串口通信的模式	Mode_Single_Line (单工) Mode_duplex (全双工)
USART_HardwareFlowControl	uint16_t	指定是启用还是禁用硬件流控制模式。	FlowCtrl_None (禁用流控) FlowCtrl_Enable (启用流控)
USART_TXLen	uint16_t	指定 Tx DMA buff len	指定 Tx DMA buff 长度
USART_RXLen	uint16_t	指定 Rx DMA buff len	指定 Rx DMA buff 长度

3 函数说明

3.1 UART 初始化函数——USART_Init

- 函数原型: void USART_Init(USART_TypeDef USARTx, USART_InitTypeDef* USART_InitStruct);

- 说明：串口初始化。

参数	方向	说明
USART_TypeDef USARTx	IN	选择 UART。该参数可设置为以下值之一： UARTA, UARTB.
USART_InitTypeDef* USART_InitStruct	IN	该指针指向内容包含所需要配置的 UART 中所有配置参数

表 3.1 USART_Init 形参表

返回值	说明
None	无返回值

表 3.2 USART_Init 返回值

3.2 关闭 UART 使能函数——USART_DeInit

- 函数原型：void USART_DeInit(USART_TypeDef USARTx)
- 说明：去除 USARTx 初始化函数，目的是恢复 USARTx 寄存器初始值

参数	方向	说明
USART_TypeDef USARTx	IN	选择 UART。该参数可设置为以下值之一： UARTA, UARTB.

表 3.3 USART_DeInit 形参表

返回值	说明
None	无返回值

表 3.4 USART_DeInit 返回值

3.3 单个数据发送函数——USART_SendData

- 函数原型：void USART_SendData(USART_TypeDef USARTx, uint8_t Data)
- 说明：通过串口 USARTx 发送单个的数据。

参数	方向	说明
USART_TypeDef USARTx	IN	选择 UART。该参数可设置为以下值之一： UARTA, UARTB.
uint8_t Data	IN	待发送的数据

表 3.5 USART_SendData 形参列表

返回值	说明
None	无返回值

表 3.6 USART_SendData 返回值

3.4 单个数据接收函数——USART_ReceiveData

- 函数原型：uint8_t USART_ReceiveData(USART_TypeDef USARTx)
- 说明：通过串口 USARTx 接收一个数据。

参数	方向	说明
USART_TypeDef USARTx	IN	选择 UART。该参数可设置为以下值之一： UARTA, UARTB.

表 3.7 USART_ReceiveData 形参列表

返回值	说明
Data	接收的一个字节的数据

表 3.8 USART_ReceiveData 返回值

3.5 获取 uart 未读数据长度——USART_GetRxCount

- 函数原型：uint16_t USART_GetRxCount(USART_TypeDef USARTx)
- 说明：获取当前 RX DMA 中剩余的未读取数据总数

参数	方向	说明
USART_TypeDef USARTx	IN	选择 UART。该参数可设置为以下值之一： UARTA, UARTB.

表 3.9 USART_GetRxCount 形参列表

返回值	说明
Length	获取 RX DMA 剩余读取个数

表 3.10 USART_GetRxCount 返回值

3.6 USART_ReadDatatoBuff

- 函数原型：uint16_t USART_ReadDatatoBuff(USART_TypeDef USARTx, uint8_t* RxBuff, uint16_t RxSize);
- 说明：将通过串口 USARTx 读取指定长度的数据保存到指定的 BUFF 当中，并返回接收数据的长度

参数	方向	说明
USART_TypeDef USARTx	IN	选择 UART。该参数可设置为以下值之一： UARTA, UARTB.
uint8_t* RxBuff	OUT	接收数据存放的首地址
uint16_t RxSize	OUT	接收数据的长度

表 3.11 USART_ReadDatatoBuff 形参列表

返回值	说明
Length	接收数据的长度

表 3.12 USART_ReadDatatoBuff 返回值

3.7 USART_SendDataFromBuff

- 函数原型：uint16_t USART_SendDataFromBuff(USART_TypeDef USARTx, uint8_t* TxBuff, uint16_t TxLen)
- 说明：通过串口 USARTx 将从指定位置开始的指定长度的数据发送出去。

参数	方向	说明
----	----	----

USART_TypeDef USARTx	IN	选择 UART。该参数可设置为以下值之一： UARTA, UARTB.
uint8_t* TxBuff	IN	待发送数据起始地址
uint16_t TxLen	IN	待发送数据的长度

表 3.13 USART_SendDataFromBuff 形参列表

返回值	说明
None	无返回值

s 表 3.14 USART_SendDataFromBuff 返回值

4 Demo 函数说明

4.1 初始化——UartxInit

串口使用时的常规配置。

```
void USARTx_Configuration(USART_TypeDef USARTx)
{
    //串口相关参数配置
    USART_InitTypeDef USART_InitStruct ;
    USART_InitStruct.USART_BaudRate = BauDate;
    USART_InitStruct.USART_HardwareFlowControl=
    USART_HardwareFlowControl_None;
    USART_InitStruct.USART_WordLength = USART_WordLength_8b;
    USART_InitStruct.USART_StopBits  = USART_StopBits_1;
    USART_InitStruct.USART_Mode = USART_Mode_duplex;
    USART_InitStruct.USART_Parity= USART_Parity_Even ;
    USART_InitStruct.USART_TXLen = 1024;
    USART_InitStruct.USART_RXLen  = 1024;

    //配置 GPIO 为串口功能
    if(UARTA == USARTx){
        GPIO_SetGpioMultiFunction((GPIO_NUM)printportcb.print_txio,
        GPCFG_UART_TXD);
        GPIO_SetGpioMultiFunction((GPIO_NUM)printportcb.print_rxio,
        GPCFG_UART_RXD);
    }else if (UARTB == USARTx){
        GPIO_SetGpioMultiFunction((GPIO_NUM)printportcb.print_txio,
        GPCFG_UARTB_TXD);
        GPIO_SetGpioMultiFunction((GPIO_NUM)printportcb.print_rxio,
        GPCFG_UARTB_RXD);
    }
    USART_Init(USARTx,&USART_InitStruct);//串口的初始化
}
}
```