

一起学习 CC3200 系列教程之跑马灯

阿汤哥

序:

能力有限，英语不怎么好，难免有错，有问题请联系我，

QQ1519256298 hytga@163.com

现在我们来学习 CC3200 的串口，实现简单的发送与接收。

分成

- (一)、CC3200 的串口库函数介绍
- (二)、硬件介绍
- (三)、软件介绍

(一)、CC3200 的串口寄存器介绍:

CC3200 的串口跟其他的单片机的串口基本一样，有两个不同点:

- 1、多了 FIFO
- 2、发送的数据寄存器和接收的数据寄存器一样

看一下 CC3200 的框图:

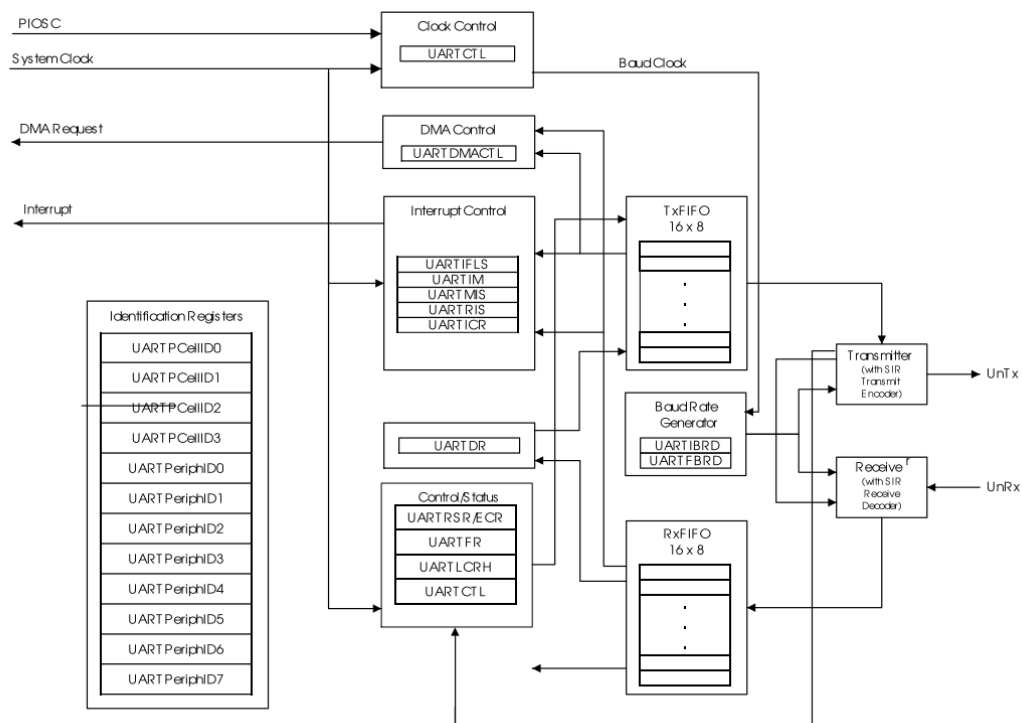


Figure 6-1. UART Module Block Diagram

从上图我们可以看到接收的数据经过 Receive 模块，FIFO 模块，最后到达 UARTDR 寄存器，发送的数据经过 UARTDR 寄存器，FIFO 模块，最后到达 transmitter 模块，

CC3200 串口设置一般有以下步骤

- 1、引脚映射
- 2、串口时钟使能
- 3、波特率及其参数设置
- 4、FIFO 设置
- 5、中断设置
- 6、中断函数编写
- 7、发送数据和接收数据

下面，我们就简单地介绍了与串口有关的库函数

- 1、 引脚映射

```
PinTypeUART(PIN_55, PIN_MODE_3);
```

- 2、 使能串口时钟

```
PRCMPeripheralClkEnable(PRCM_UARTA0, PRCM_RUN_MODE_CLK);
```

- 3、 波特率及其参数设置

```
UARTConfigSetExpClk(UARTA0_BASE, PRCMPeripheralClockGet(PRCM_UARTA0), 115200,  
(UART_CONFIG_WLEN_8|UART_CONFIG_STOP_ONE|UART_CONFIG_PAR_NONE));
```

UARTA0_BASE: 串口 0

PRCMPeripheralClockGet(PRCM_UARTA0): 获取串口的时钟频率

115200: 波特率

(UART_CONFIG_WLEN_8|UART_CONFIG_STOP_ONE|UART_CONFIG_PAR_NONE):

8bit 的数据位, 1bit 的停止位, 没有奇偶校验位

- 4、 FIFO 设置

CC3200 有发送数据的 FIFO 也有接收数据的 FIFO, 一般的应用都是发送不用触发中断, 而接收需要触发中断,

```
UARTFIFODisable(UARTA0_BASE); //失能串口 0 的 FIFO 功能, 当设置接收触发中断后, 每  
接收到一个字符就触发一个中断,
```

```
UARTFIFOLevelSet(UARTA0_BASE, UART_FIFO_TX1_8, UART_FIFO_RX1_8); //设置 FIFO 的触发  
阈值, CC3200 的 FIFO 是 16 个 8bit 的寄存器, 当设置接收触发中断, UART_FIFO_RX1_8  
设置的阈值是 2 字节, 当接收到 2 个以上的字节就会触发中断。
```

- 5、 中断设置

```
UARTIntRegister(UARTA0_BASE, uart_handle); //注册一个中断回调函数,
```

UARTA0_BASE: 串口 0

uart_handle: 我们的中断回调函数, 其声明为 `uartvoid uart_handle(void)`

```
UARTIntEnable(UARTA0_BASE, UART_INT_RX); //使能接收中断
```

- 6、 中断函数编写

首先获取串口中断状态寄存器的值 (谁触发的):

```
UARTIntStatus(UARTA0_BASE, true);
```

CC3200 的串口中断状态寄存器有两种:

- 1、 没有经过屏蔽的中断状态寄存器
- 2、 经过屏蔽的中断状态寄存器

先介绍一下中断触发的机制, 任意的中断触发源经过一个中断屏蔽寄存器, 再触发中断。没有经过屏蔽的中断状态寄存器记录的是所有的中断触发源 (可能有很多的触发源触发了中

断，但是不一定产生中断)，经过屏蔽的中断状态寄存器记录的是触发这次中断的触发源（一定是这个触发源产生中断的）

True: 获取经过屏蔽的中断状态寄存器，一般应用用的是这个，毕竟这个才是我们关心

UARTCharsAvail(UARTAO_BASE), 参看接收的 FIFO 有没有数据

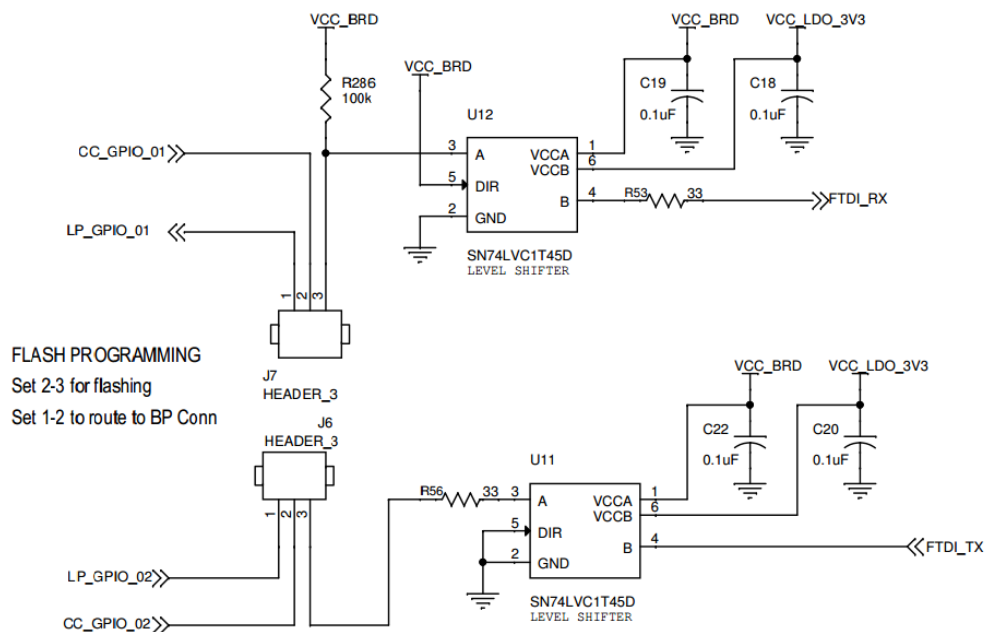
UARTIntClear(UARTAO_BASE,UART_INT_RX);清楚中断标志位

7、

UARTCharGet(UARTAO_BASE); //获取接收数据

UARTCharPut(UARTAO_BASE,'a');//发送数据

(二)、硬件介绍



可以看到 GPIO_02 是发送 RXD 对应的是 FTDI_TX，GPIO_01 是 TXD 对应的是 FTDI_RX

Table 16-7. Pin Multiplexing (continued)

General Pin Attributes						Function		
Pkg Pin	Pin Alias	Use	Select as Wakeup Source	Config Addl Analog Mux	Muxed with JTAG	Dig. Pin Mux Config Reg	Dig. Pin Mux Config Mode Value	Signal Name
55	GPIO1	I/O	No	No	No	GPIO_PAD_CONFIG_1 (0x4402 E0A4)	0	GPIO1
							3	UART0_TX
							4	pCLK (PIXCLK)
							6	UART1_TX
							7	GT_CCP01
56	VDD_DIG2	Internal Power	N/A	N/A	N/A	N/A		VDD_DIG2
57 ⁽¹²⁾	GPIO2	Analog Input (up to 1.5 V)/ Digital I/O	Wake-Up Source	See ⁽¹³⁾ (¹⁴)	No	GPIO_PAD_CONFIG_2 (0x4402 E0A8)	See ⁽¹⁵⁾	ADC_CH0
							0	GPIO2
							3	UART0_RX
							6	UART1_RX
							7	GT_CCP02

(三)、软件介绍

软件功能：简答的回显例程，关闭 FIFO 功能

```
void uart_handle(void) {
    int i,k,j;
    #if 1
        //获取经过屏蔽的中断状态
        i = UARTIntStatus(UARTA0_BASE,true);
        //接收中断
        if(i&UART_INT_RX) {

            //判断FIFO有没有数据
            while(UARTCharsAvail(UARTA0_BASE)){
                //获取接收的数据，并直接再发送
                UARTCharPut(UARTA0_BASE,UARTCharGet(UARTA0_BASE));
            }
            //清除中断标志位
            UARTIntClear(UARTA0_BASE,UART_INT_RX);

        }
    #endif

}

void uart0_Init(void) {
    // 使能时钟
    PRCMPeripheralClkEnable(PRCM_UARTA0, PRCM_RUN_MODE_CLK);
```

```

//引脚映射
PinTypeUART(PIN_55, PIN_MODE_3);
PinTypeUART(PIN_57, PIN_MODE_3);
//波特率及其参数设置
UARTConfigSetExpClk(UARTA0_BASE,PRCMPeripheralClockGet(PRCM_UARTA0),
                    UART_BAUD_RATE, (UART_CONFIG_WLEN_8 | UART_CONFIG_STOP_ONE |
                    UART_CONFIG_PAR_NONE));
//设置中断回调函数
UARTIntRegister(UARTA0_BASE,uart_handle);
//设置接收中断
UARTIntEnable(UARTA0_BASE,UART_INT_RX);

//失能fifo
UARTFIFODisable(UARTA0_BASE);

//设置FIFO的阈值
//UARTFIFOLevelSet(UARTA0_BASE,UART_FIFO_TX1_8,UART_FIFO_RX1_8);
//发送一个a
UARTCharPut(UARTA0_BASE,'a');
}
void main()
{
    char cString[MAX_STRING_LENGTH+1];
    char cCharacter;
    int iStringLength = 0;
    //
    // Initailizing the board
    //
    BoardInit();
    uart0_Init();
    while(1){};
}

```