

1. 将demo目录\sdk\platform\driver下的uart2文件复制到客户工程对应的目录下。
2. 将demo目录\sdk\platform\reg下的reg_uart2.h文件复制到客户工程对应的文件，如果客户工程对应的目录已经有reg_uart2.h文件，则直接替换。
3. 在BK3435_reg.h添加 宏定义

```
#define REG_APB3_UART2_CFG          (*((volatile unsigned long *) 0x00806A00))
#define REG_APB3_UART2_FIFO_CFG     (*((volatile unsigned long *) 0x00806A04))
#define REG_APB3_UART2_FIFO_STAT    (*((volatile unsigned long *) 0x00806A08))
#define REG_APB3_UART2_PORT         (*((volatile unsigned long *) 0x00806A0C))
#define REG_APB3_UART2_INT_ENABLE   (*((volatile unsigned long *) 0x00806A10))
#define REG_APB3_UART2_INT_STAT     (*((volatile unsigned long *) 0x00806A14))
#define REG_APB3_UART2_FLOW_CFG     (*((volatile unsigned long *) 0x00806A18))
#define REG_APB3_UART2_WAKE_CFG     (*((volatile unsigned long *) 0x00806A1C))
13: // UART
14: //-----
15: #if UART_ONE_ENBLE
16: #define REG_APB3_UART_CFG          (*((volatile unsigned long *) 0x00806300))
17: #define REG_APB3_UART_FIFO_CFG     (*((volatile unsigned long *) 0x00806304))
18: #define REG_APB3_UART_FIFO_STAT    (*((volatile unsigned long *) 0x00806308))
19: #define REG_APB3_UART_PORT         (*((volatile unsigned long *) 0x0080630C))
20: #define REG_APB3_UART_INT_ENABLE   (*((volatile unsigned long *) 0x00806310))
21: #define REG_APB3_UART_INT_STAT     (*((volatile unsigned long *) 0x00806314))
22: #define REG_APB3_UART_FLOW_CFG     (*((volatile unsigned long *) 0x00806318))
23: #define REG_APB3_UART_WAKE_CFG     (*((volatile unsigned long *) 0x0080631C))
24: #endif
25:
26: #if 1//UART_ONE_ENBLE
27: #define REG_APB3_UART_CFG          (*((volatile unsigned long *) 0x00806A00))
28: #define REG_APB3_UART_FIFO_CFG     (*((volatile unsigned long *) 0x00806A04))
29: #define REG_APB3_UART_FIFO_STAT    (*((volatile unsigned long *) 0x00806A08))
30: #define REG_APB3_UART_PORT         (*((volatile unsigned long *) 0x00806A0C))
31: #define REG_APB3_UART_INT_ENABLE   (*((volatile unsigned long *) 0x00806A10))
32: #define REG_APB3_UART_INT_STAT     (*((volatile unsigned long *) 0x00806A14))
33: #define REG_APB3_UART_FLOW_CFG     (*((volatile unsigned long *) 0x00806A18))
34: #define REG_APB3_UART_WAKE_CFG     (*((volatile unsigned long *) 0x00806A1C))
35: #endif
36:
37: #define INT_STATUS_UART2_bit        (0x01<<16)
38: #define INT_STATUS_ADC_bit         (0x01<< 8)
39: #define INT_STATUS_I2C_bit         (0x01<< 7)
40: #define INT_STATUS_SPI_bit         (0x01<< 6)
41: #if UART_ONE_ENBLE
42: #define INT_STATUS_UART_bit        (0x01<< 5)
43: #endif
44: #define INT_STATUS_UART2_bit        (0x01<<16)
45:
46: #define INT_STATUS_PWM4_bit         (0x01<< 4)
47: #define INT_STATUS_PWM3_bit         (0x01<< 3)
```

4. 在int.h文件添加宏定义

```
#define INT_STATUS_UART2_bit        (0x01<<16)
#define INT_UART2_bit              (0x01<<16)
48: #define INT_STATUS_I2C_bit         (0x01<< 7)
49: #define INT_STATUS_SPI_bit         (0x01<< 6)
50: #if UART_ONE_ENBLE
51: #define INT_STATUS_UART_bit        (0x01<< 5)
52: #endif
53: #define INT_STATUS_UART2_bit        (0x01<<16)
54:
55: #define INT_STATUS_PWM4_bit         (0x01<< 4)
56: #define INT_STATUS_PWM3_bit         (0x01<< 3)
57: #if UART_ONE_ENBLE
58: #define INT_UART_bit              (0x01<< 5)
59: #endif
60: #define INT_UART2_bit              (0x01<<16)
61:
62: #define INT_PWM4_bit              (0x01<< 4)
63: #define INT_PWM3_bit              (0x01<< 3)
```

5. 在int.c文件修改void intc_init(void)函数

```

void intc_init(void)
{
    // Clear all interrupts
    intc_enable_clear(INT_IRQ_BIT | FIQ_IRQ_BIT);

    // enable the supported interrupts
    intc_enable_set(INT_IRQ_BIT | FIQ_IRQ_BIT);

    intc_module_enable_set(INT_BLE_bit | INT_UART_bit | INT_UART2_bit);
}

```

在void IRQ_Exception(void)函数调用中断函数uart2_isr();

```

}
#ifdef (UART_DRIVER)
// call the function handler
if(IntStat & INT_STATUS_UART_bit)
{
    irq_status |= INT_STATUS_UART_bit;
    uart_isr();
}
#endif
#ifdef (UART_DRIVER)
// call the function handler
if(IntStat & INT_STATUS_UART2_bit)
{
    irq_status |= INT_STATUS_UART2_bit;
    uart2_isr();
}
#endif

```

6. 在arch_main.c文件rw_main()函数里调用uart2_init(115200);初始化函数和uart2_cb_register(uart2_rx_handler);函数

```

84: intc_init();
85: // Initialize UART component
86: #if (UART_DRIVER)
87: uart_init(115200);
88: uart_cb_register(uart_rx_handler);
89: uart2_init(115200);
90: uart2_cb_register(uart2_rx_handler); //uart2.
91: #endif
92: uart_stack_register(uart_printf);
93: flash_advance_init();

```

7. 增加static void uart2_rx_handler(uint8_t *buf, uint8_t len)并且声明

```

#ifdef (UART_DRIVER)
static void uart_rx_handler(uint8_t *buf, uint8_t len)
{
    for(uint8_t i=0; i<len; i++)
    {
        UART_PRINTF("0x%x ", buf[i]);
    }
    UART_PRINTF("\r\n");
}

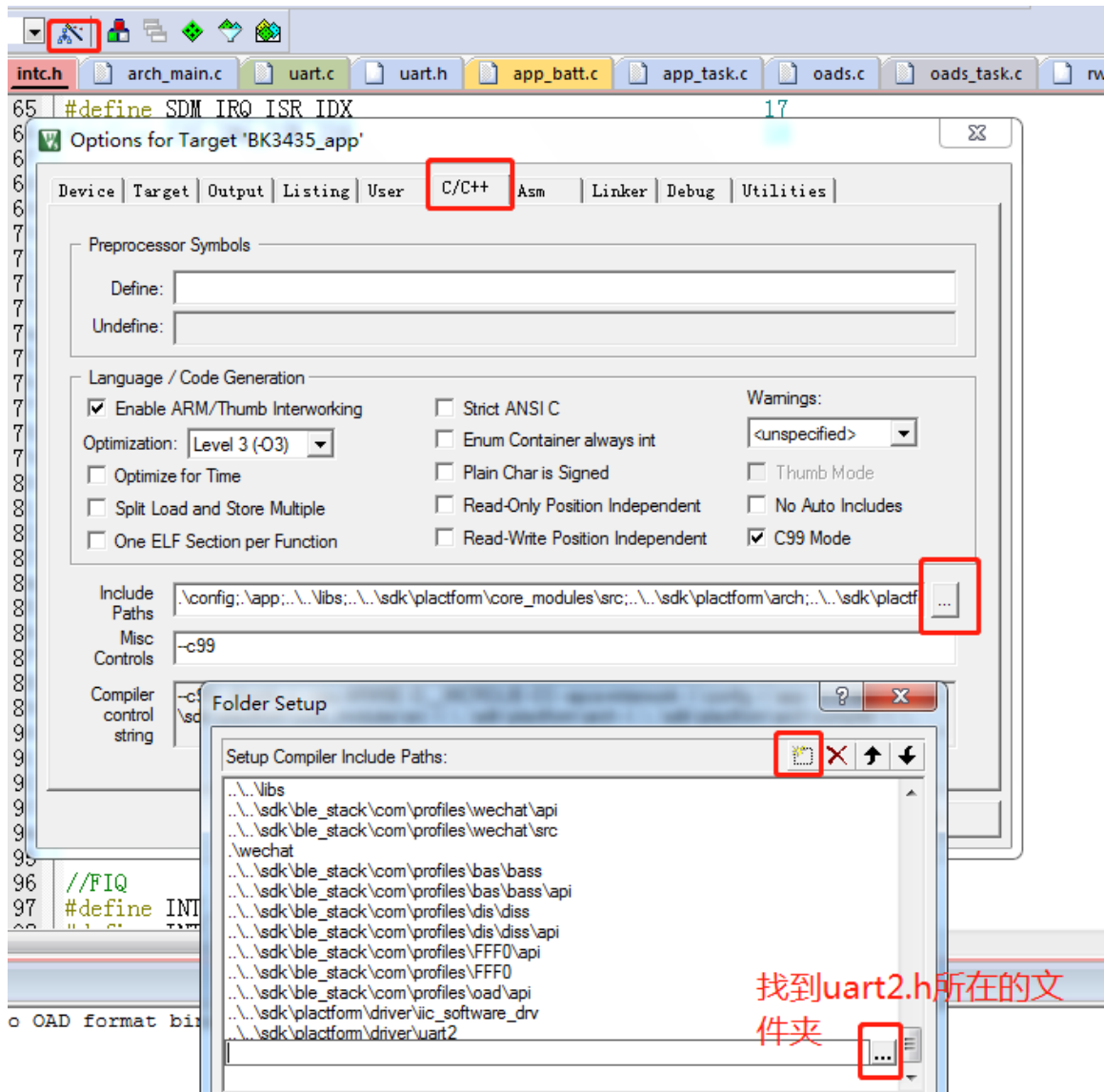
static void uart2_rx_handler(uint8_t *buf, uint8_t len) //uart2.
{
    for(uint8_t i=0; i<len; i++)
    {
        Uart2_printf("0x%x ", buf[i]);
    }
    Uart2_printf("\r\n");
}

#endif

#ifdef (UART_DRIVER)
void uart_rx_handler(uint8_t *buf, uint8_t len);
void uart2_rx_handler(uint8_t *buf, uint8_t len); //uart2. 函数声明
#endif

```

- 在相应的文件下调用#include "uart2.h"头文件声明。注意P17、P16不要做任何配置。
- 在keil工程里添加uart2.c和uart2.h文件。



- 在arch_main.c文件rw_main()函数里调用Uart2_printf("Uart2_printf Mode Start\r\n");UART2打印函数。

```
else //normal mode
{
    UART_PRINTF("NORMAL Mode Start\r\n");
    Uart2_printf("Uart2_printf Mode Start\r\n");
    rw_app_enter();
}
```

- 打开串口助手，串口助手发送数据后，芯片会通过UART2将数据原样返回。

