

BLE_UART SDK 测试说明

一、软件

- 1、UART 的 TX、RX 初始化为 P2、P3 脚，波特率为 115200（如下图）。

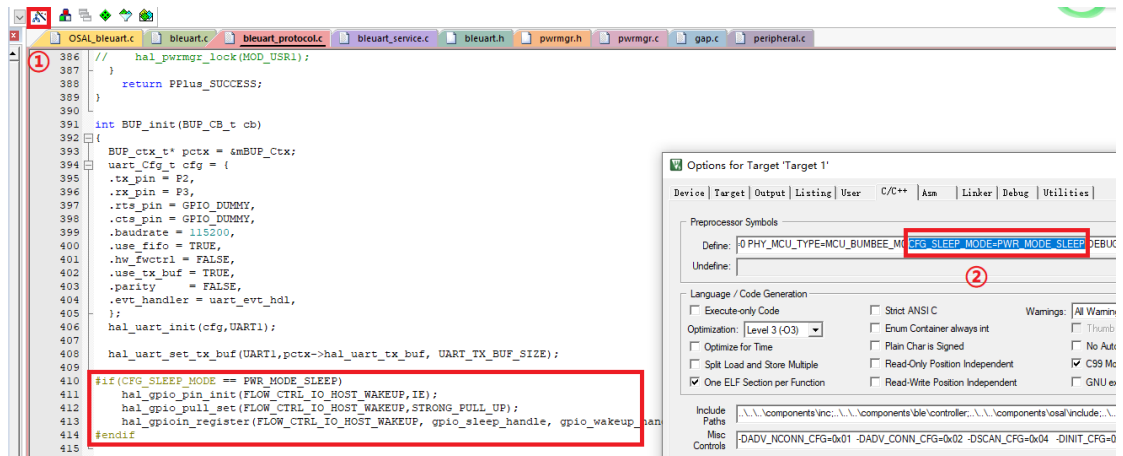
若需修改初始化引脚、波特率等可在 BUP_init 函数中修改。

```
int BUP_init(BUP_CB_t cb)
{
    BUP_ctx_t* pctx = &mBUP_Ctx;
    uart_Cfg_t cfg = {
        .tx_pin = P2,
        .rx_pin = P3,
        .rts_pin = GPIO_DUMMY,
        .cts_pin = GPIO_DUMMY,
        .baudrate = 115200,
        .use_fifo = TRUE,
        .hw_fwctrl = FALSE,
        .use_tx_buf = TRUE,
        .parity = FALSE,
        .evt_handler = uart_evt_hdl,
    };
    hal_uart_init(cfg, UART1);

    hal_uart_set_tx_buf(UART1, pctx->hal_uart_tx_buf, UART_TX_BUF_SIZE);

#ifdef CFG_SLEEP_MODE == PWR_MODE_SLEEP
    hal_gpio_pin_init(FLOW_CTRL_IO_HOST_WAKEUP, IE);
    hal_gpio_pull_set(FLOW_CTRL_IO_HOST_WAKEUP, STRONG_PULL_UP);
    hal_gpioin_register(FLOW_CTRL_IO_HOST_WAKEUP, gpio_sleep_handle, gpio_wakeup_handle);
#endif
}
```

- 2、可以通过下图的①、②步骤设定宏 CFG_SLEEP_MODE=PWR_MODE_SLEEP 允许系统进入休眠模式或 CFG_SLEEP_MODE=PWR_MODE_NO_SLEEP 不允许系统进入休眠模式。



```

#if(CFG_SLEEP_MODE == PWR_MODE_SLEEP)
    hal_gpio_pin_init(FLOW_CTRL_IO_HOST_WAKEUP, IE);
    hal_gpio_pull_set(FLOW_CTRL_IO_HOST_WAKEUP, STRONG_PULL_UP);
    hal_gpiopin_register(FLOW_CTRL_IO_HOST_WAKEUP, gpio_sleep_handle, gpio_wakeup_handle);
#endif

```

当允许系统进入休眠模式时，UART 会进入休眠模式，此时若要从 UART 发送数据到 BLE 需要配置唤醒脚（如上图）上拉；当需要 UART 发送数据时需将唤醒脚接地，即将 FLOW_CTRL_IO_HOST_WAKEUP 接地从而将 UART 从休眠模式唤醒。唤醒脚为 P14，若需要修改可通过宏定义 FLOW_CTRL_IO_HOST_WAKEUP 修改唤醒脚，如下图：

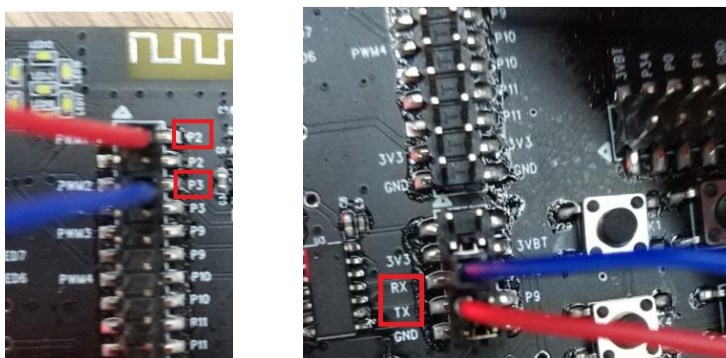
```

#define FLOW_CTRL_IO_HOST_WAKEUP    P14

```

二、硬件

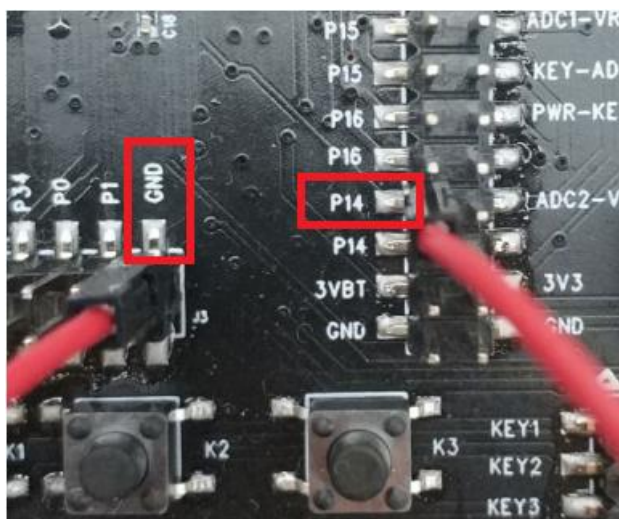
- 1、若 UART 的 TX、RX 初始化引脚未修改，且使用伦茨开发板做验证则接线如下图：
P2<——>TX、P3<——>RX



若未使用开发板，则需将 P2 接到串口模块的 RX，P3 接到串口模块的 TX。

注：测试时 IC 需供电，串口模块需共地。

- 2、若 CFG_SLEEP_MODE=PWR_MODE_SLEEP 即允许系统进入休眠模式，当需要从串口发送数据时需要将唤醒脚（若未修改 SDK 则为 P14）接地，开发板如下图：

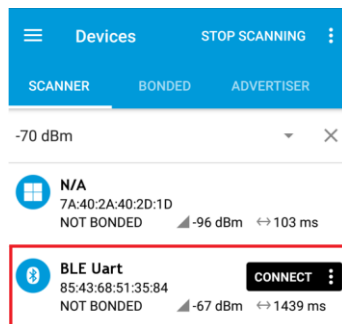


三、APP

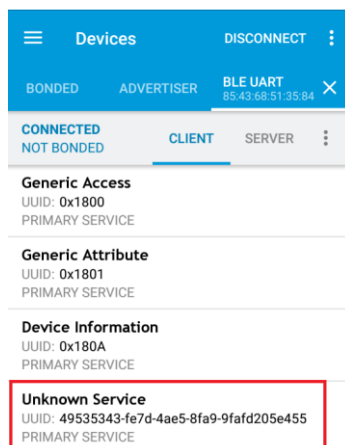
1、测试 APP 可使用 nRF Connect



2、扫描设备



3、连接设备



4、传输数据

