enum POWER\_MODE

{

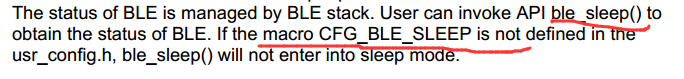
PM\_ACTIVE, /\*!< CO\_PD\_DISALLOWED, disallow cpu clock off & cpu power down \*/

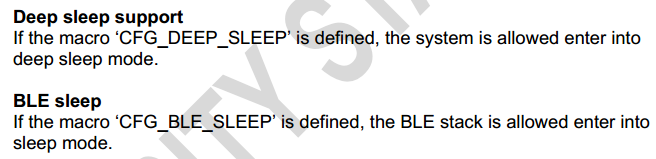
PM\_IDLE, /\*!< CPU\_CLK\_OFF\_ALLOW \*/

PM\_SLEEP, /\*!< CPU\_POWER\_DOWN\_ALLOW \*/

PM\_DEEP\_SLEEP /\*!< CPU\_DEEP\_SLEEP\_ALLOW \*/

};





调试总结：

如果：

/// Deep sleep support

#define CFG\_DEEP\_SLEEP // wenxue

///// BLE Sleep

#define CFG\_BLE\_SLEEP // Wenxue

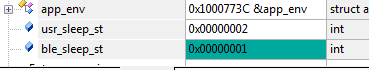
那么：



这样会进入DEEP SLEEP状态

按下按键后，进入广播状态：





连接之后：





如果这时按住button1：



鹏飞原来的工程：

#define QN\_DEMO\_AUTO 1

//#define CFG\_DEEP\_SLEEP

///// BLE Sleep

//#define CFG\_BLE\_SLEEP

这时上电后自动开始广播，

usr\_sleep\_st = usr\_sleep(); 得到的结果是PM\_SLEEP

ble\_sleep\_st = ble\_sleep(usr\_sleep\_st);得到的结果是 PM\_IDLE

进入的状态是enter\_sleep(SLEEP\_CPU\_CLK\_OFF,

WAKEUP\_BY\_ALL\_IRQ\_SOURCE,

NULL);

进入低功耗后：

#define QN\_DEMO\_AUTO 0 // Wenxue

这时通过按键按下一次之后开始广播

/// Deep sleep support

#define CFG\_DEEP\_SLEEP // wenxue

///// BLE Sleep

#define CFG\_BLE\_SLEEP // Wenxue

打开，允许进入低功耗状态

下面这个非常重要：

void usr\_sleep\_restore(void)

{

#if QN\_DBG\_PRINT

uart\_init(QN\_DEBUG\_UART, USARTx\_CLK(0), UART\_115200);

uart\_tx\_enable(QN\_DEBUG\_UART, MASK\_ENABLE);

uart\_rx\_enable(QN\_DEBUG\_UART, MASK\_ENABLE);

#endif

// wenxue 20160916

uart\_init(QN\_UART0, USARTx\_CLK(0), UART\_115200);

uart\_tx\_enable(QN\_UART0, MASK\_ENABLE);

uart\_rx\_enable(QN\_UART0, MASK\_ENABLE);

pt\_enable(); // uart data transfer enable

}