AIC8800D
Wi-Fi6/BT5.0 SoC
休眠唤醒调适手册

Revision: 1.0

2022/06/01

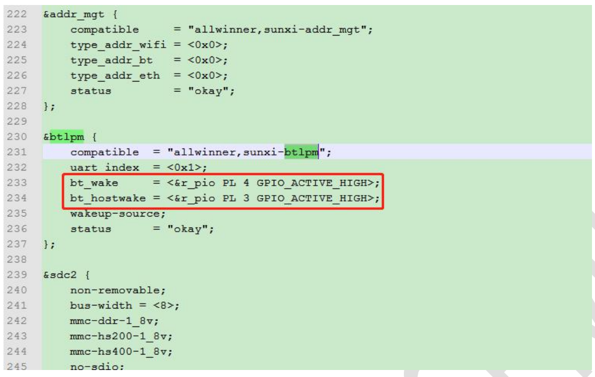
历史更新记录

时间	修改内容	修订人	版本
2022/06/01	初版	Aiden	1.0

全志 Allwinner 移植方式(SDIO)

dts 部分

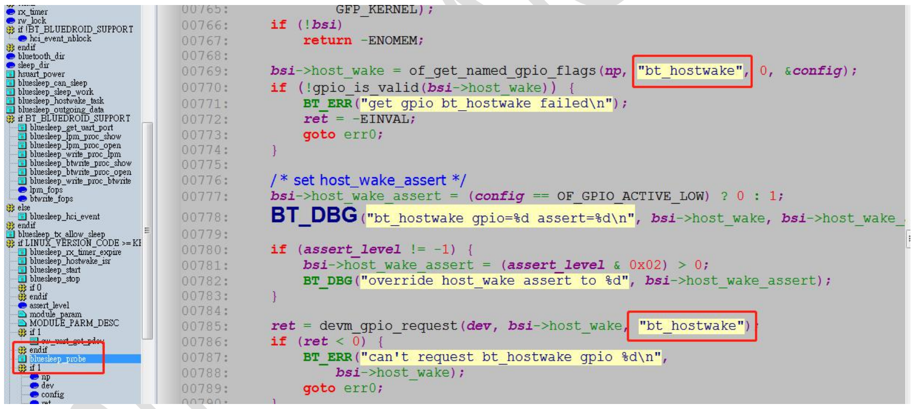
需要确认 bt_wake 和 bt_host_wake 是否设定在 dts 当中
如 A133 为例，可在
android/longan/device/config/chips/a133/configs/b4/linux-5.4/board.dts
中看到该设定



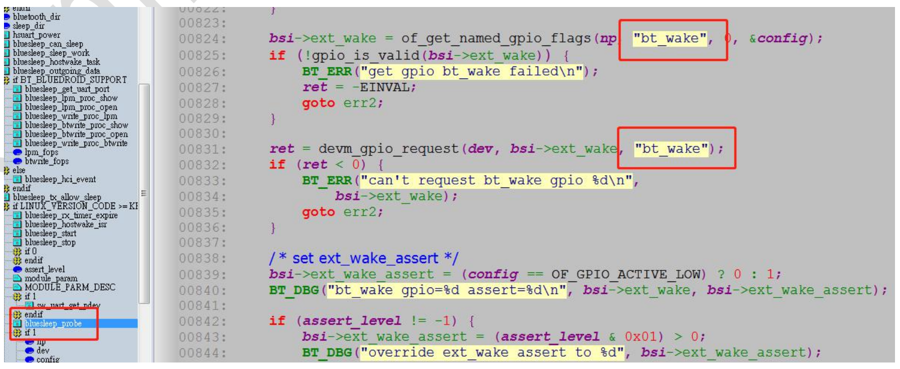
```
&addr_mgt {
    compatible     = "allwinner,sunxi-addr_mgt";
    type_addr_wifi = <0x0>;
    type_addr_bt   = <0x0>;
    type_addr_eth  = <0x0>;
    status         = "okay";
};

&btlpm {
    compatible  = "allwinner,sunxi-btlpm";
    uart_index  = <0x1>;
    bt_wake     = <&r_pio PL 4 GPIO_ACTIVE_HIGH>;
    bt_hostwake = <&r_pio PL 3 GPIO_ACTIVE_HIGH>;
    wakeup-source;
    status      = "okay";
};

&sdc2 {
    non-removable;
    bus-width = <8>;
    mmc-ddr-1_8v;
    mmc-hs200-1_8v;
    mmc-hs400-1_8v;
    no-sdio;
```

aic8800_btlpm

将定义好的 bt_wake 和 bt_host_wake 设定在 aic8800_btlpm 中



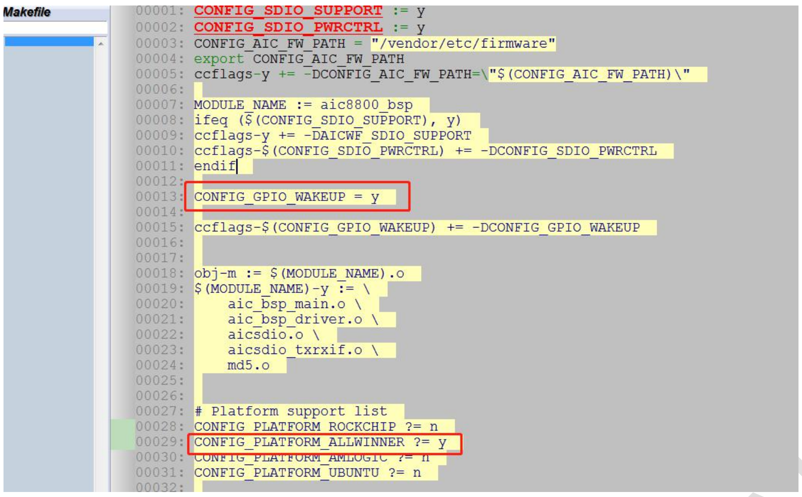
```
            GFP_KERNEL);
    if (!bsi)
        return -ENOMEM;

    bsi->host_wake = of_get_named_gpio_flags(np, "bt_hostwake", 0, &config);
    if (!gpio_is_valid(bsi->host_wake)) {
        BT_ERR("get gpio bt_hostwake failed\n");
        ret = -EINVAL;
        goto err0;
    }

    /* set host_wake_assert */
    bsi->host_wake_assert = (config == OF_GPIO_ACTIVE_LOW) ? 0 : 1;
    BT_DBG("bt_hostwake gpio=%d assert=%d\n", bsi->host_wake, bsi->host_wake_

    if (assert_level != -1) {
        bsi->host_wake_assert = (assert_level & 0x02) > 0;
        BT_DBG("override host_wake assert to %d", bsi->host_wake_assert);
    }

    ret = devm_gpio_request(dev, bsi->host_wake, "bt_hostwake");
    if (ret < 0) {
        BT_ERR("can't request bt_hostwake gpio %d\n",
            bsi->host_wake);
        goto err0;
```



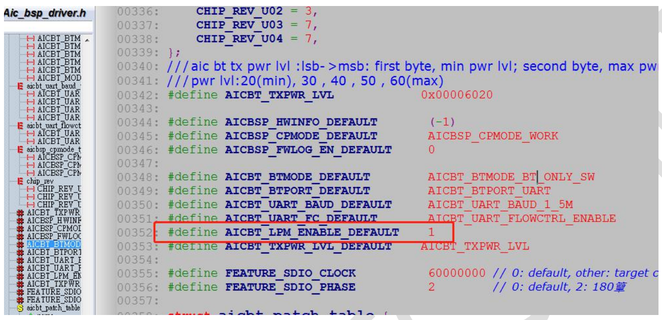
```
    }

    bsi->ext_wake = of_get_named_gpio_flags(np, "bt_wake", 0, &config);
    if (!gpio_is_valid(bsi->ext_wake)) {
        BT_ERR("get gpio bt_wake failed\n");
        ret = -EINVAL;
        goto err2;
    }

    ret = devm_gpio_request(dev, bsi->ext_wake, "bt_wake");
    if (ret < 0) {
        BT_ERR("can't request bt_wake gpio %d\n",
            bsi->ext_wake);
        goto err2;
    }

    /* set ext_wake_assert */
    bsi->ext_wake_assert = (config == OF_GPIO_ACTIVE_LOW) ? 0 : 1;
    BT_DBG("bt_wake gpio=%d assert=%d\n", bsi->ext_wake, bsi->ext_wake_assert);

    if (assert_level != -1) {
        bsi->ext_wake_assert = (assert_level & 0x01) > 0;
        BT_DBG("override ext_wake assert to %d", bsi->ext_wake_assert);
```

aic8800_bsp

Makefile 中将以下两个 CONFIG 改成 y，并将在 aic_bsp_driver.h 中将
AICBT_LPM_ENABLE_DEFAULT 改成 1



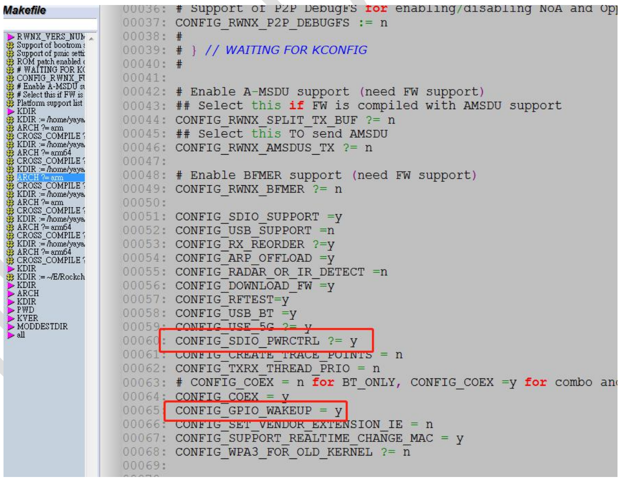
```
CONFIG_SDIO_SUPPORT := y
CONFIG_SDIO_PWRCTRL := y
CONFIG_AIC_FW_PATH = "/vendor/etc/firmware"
export CONFIG_AIC_FW_PATH
ccflags-y += -DCONFIG_AIC_FW_PATH=\"$(CONFIG_AIC_FW_PATH)\"

MODULE_NAME := aic8800_bsp
ifeq ($(CONFIG_SDIO_SUPPORT), y)
ccflags-y += -DAICWF_SDIO_SUPPORT
ccflags-$(CONFIG_SDIO_PWRCTRL) += -DCONFIG_SDIO_PWRCTRL
endif

CONFIG_GPIO_WAKEUP = y

ccflags-$(CONFIG_GPIO_WAKEUP) += -DCONFIG_GPIO_WAKEUP


obj-m := $(MODULE_NAME).o
$(MODULE_NAME)-y := \
    aic_bsp_main.o \
    aic_bsp_driver.o \
    aicsdio.o \
    aicsdio_txrxif.o \
    md5.o


# Platform support list
CONFIG_PLATFORM_ROCKCHIP ?= n
CONFIG_PLATFORM_ALLWINNER ?= y
CONFIG_PLATFORM_AMLOGIC ?= n
CONFIG_PLATFORM_UBUNTU ?= n

```



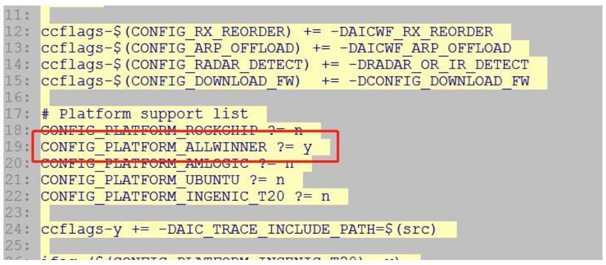
```
        CHIP_REV_U02 = 3,
        CHIP_REV_U03 = 7,
        CHIP_REV_U04 = 7,
};
///aic bt tx pwr lvl :lsb->msb: first byte, min pwr lvl; second byte, max pw
///pwr lvl:20(min), 30 , 40 , 50 , 60(max)
#define AICBT_TXPWR_LVL                 0x00006020

#define AICBSP_HWINFO_DEFAULT            (-1)
#define AICBSP_CPMODE_DEFAULT            AICBSP_CPMODE_WORK
#define AICBSP_FWLOG_EN_DEFAULT          0

#define AICBT_BTMODE_DEFAULT             AICBT_BTMODE_BT_ONLY_SW
#define AICBT_BTPORT_DEFAULT             AICBT_BTPORT_UART
#define AICBT_UART_BAUD_DEFAULT          AICBT_UART_BAUD_1_5M
#define AICBT_UART_FC_DEFAULT            AICBT_UART_FLOWCTRL_ENABLE
#define AICBT_LPM_ENABLE_DEFAULT         1
#define AICBT_TXPWR_LVL_DEFAULT         AICBT_TXPWR_LVL

#define FEATURE_SDIO_CLOCK               60000000 // 0: default, other: target c
#define FEATURE_SDIO_PHASE               2        // 0: default, 2: 180篁

struct aicbt_patch_table {
```

aic8800_fdrv

在 Makefile 中将以下两个 CONFIG 改成 y，平台选择 CONFIG_PLATFORM_ALLWINNER

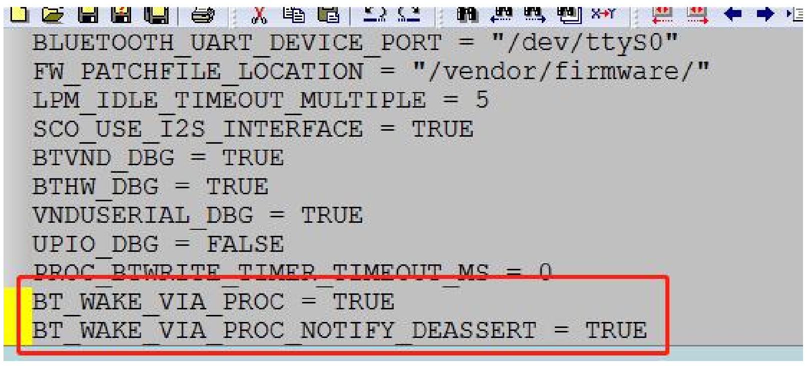


```
# Support of P2P DebugFS for enabling/disabling NoA and Op
CONFIG_RWNX_P2P_DEBUGFS := n
#
# } // WAITING FOR KCONFIG
#

# Enable A-MSDU support (need FW support)
## Select this if FW is compiled with AMSDU support
CONFIG_RWNX_SPLIT_TX_BUF ?= n
## Select this TO send AMSDU
CONFIG_RWNX_AMSDUS_TX ?= n

# Enable BFMER support (need FW support)
CONFIG_RWNX_BFMER ?= n

CONFIG_SDIO_SUPPORT =y
CONFIG_USB_SUPPORT =n
CONFIG_RX_REORDER ?=y
CONFIG_ARP_OFFLOAD =y
CONFIG_RADAR_OR_IR_DETECT =n
CONFIG_DOWNLOAD_FW =y
CONFIG_RFTEST=y
CONFIG_USB_BT =y
CONFIG_USE_5G ?= y
CONFIG_SDIO_PWRCTRL ?= y
CONFIG_CREATE_TRACE_POINTS = n
CONFIG_TXRX_THREAD_PRIO = n
# CONFIG_COEX = n for BT_ONLY, CONFIG_COEX =y for combo and
CONFIG_COEX = y
CONFIG_GPIO_WAKEUP = y
CONFIG_SET_VENDOR_EXTENSION_IE = n
CONFIG_SUPPORT_REALTIME_CHANGE_MAC = y
CONFIG_WPA3_FOR_OLD_KERNEL ?= n

```



```

ccflags-$(CONFIG_RX_REORDER) += -DAICWF_RX_REORDER
ccflags-$(CONFIG_ARP_OFFLOAD) += -DAICWF_ARP_OFFLOAD
ccflags-$(CONFIG_RADAR_DETECT) += -DRADAR_OR_IR_DETECT
ccflags-$(CONFIG_DOWNLOAD_FW)  += -DCONFIG_DOWNLOAD_FW

# Platform support list
CONFIG_PLATFORM_ROCKCHIP ?= n
CONFIG_PLATFORM_ALLWINNER ?= y
CONFIG_PLATFORM_AMLOGIC ?= n
CONFIG_PLATFORM_UBUNTU ?= n
CONFIG_PLATFORM_INGENIC_T20 ?= n

ccflags-y += -DAIC_TRACE_INCLUDE_PATH=$(src)

```

libbt-vendor



```
BLUETOOTH_UART_DEVICE_PORT = "/dev/ttyS0"
FW_PATCHFILE_LOCATION = "/vendor/firmware/"
LPM_IDLE_TIMEOUT_MULTIPLE = 5
SCO_USE_I2S_INTERFACE = TRUE
BTVND_DBG = TRUE
BTHW_DBG = TRUE
VNDUSERIAL_DBG = TRUE
UPIO_DBG = FALSE
PROC_BTWRITE_TIMER_TIMEOUT_MS = 0
BT_WAKE_VIA_PROC = TRUE
BT_WAKE_VIA_PROC_NOTIFY_DEASSERT = TRUE
```

以上设定完后即可开启休眠唤醒的功能

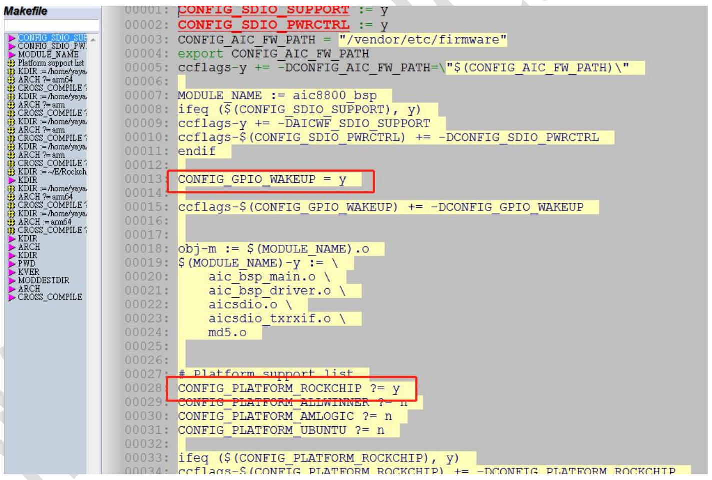
瑞芯微 Rockchip 移植方式(SDIO)

aic8800_btlpm

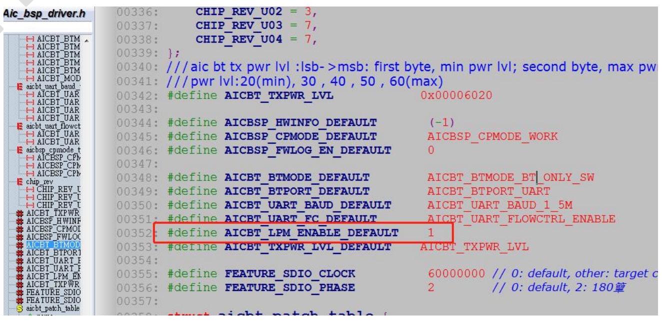
RK 平台默认已经有 bwrite 等接口，不需要使用 aic8800_btlpm 处理蓝牙休眠唤醒。

aic8800_bsp

Makefile 中将以下两个 CONFIG 改成 y，在 aic_bsp_driver.h 中将 AICBT_LPM_ENABLE_DEFAULT 改成 1



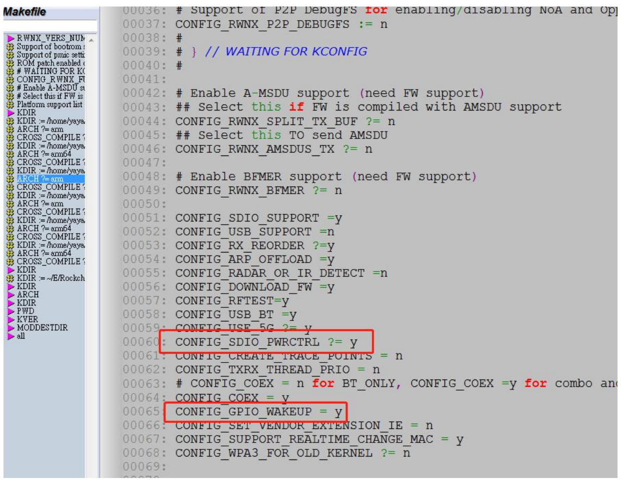
```
CONFIG_SDIO_SUPPORT := y
CONFIG_SDIO_PWRCTRL := y
CONFIG_AIC_FW_PATH = "/vendor/etc/firmware"
export CONFIG_AIC_FW_PATH
ccflags-y += -DCONFIG_AIC_FW_PATH=\"$(CONFIG_AIC_FW_PATH)\"

MODULE_NAME := aic8800_bsp
ifeq ($(CONFIG_SDIO_SUPPORT), y)
ccflags-y += -DAICWF_SDIO_SUPPORT
ccflags-$(CONFIG_SDIO_PWRCTRL) += -DCONFIG_SDIO_PWRCTRL
endif

CONFIG_GPIO_WAKEUP = y

ccflags-$(CONFIG_GPIO_WAKEUP) += -DCONFIG_GPIO_WAKEUP


obj-m := $(MODULE_NAME).o
$(MODULE_NAME)-y := \
    aic_bsp_main.o \
    aic_bsp_driver.o \
    aicsdio.o \
    aicsdio_txrxif.o \
    md5.o


# Platform support list
CONFIG_PLATFORM_ROCKCHIP ?= y
CONFIG_PLATFORM_ALLWINNER ?= n
CONFIG_PLATFORM_AMLOGIC ?= n
CONFIG_PLATFORM_UBUNTU ?= n

ifeq ($(CONFIG_PLATFORM_ROCKCHIP), y)
ccflags-$(CONFIG_PLATFORM_ROCKCHIP) += -DCONFIG_PLATFORM_ROCKCHIP
```



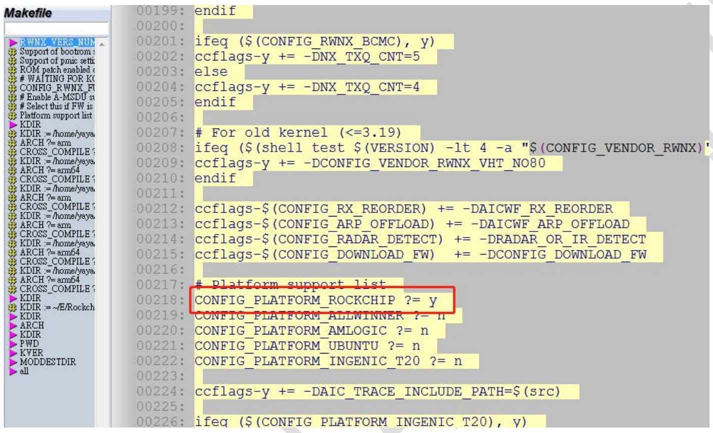
```
    CHIP_REV_U02 = 3,
    CHIP_REV_U03 = 7,
    CHIP_REV_U04 = 7,
};
///aic bt tx pwr lvl :lsb->msb: first byte, min pwr lvl; second byte, max pw
///pwr lvl:20(min), 30 , 40 , 50 , 60(max)
#define AICBT_TXPWR_LVL              0x00006020

#define AICBSP_HWINFO_DEFAULT         (-1)
#define AICBSP_CPMODE_DEFAULT         AICBSP_CPMODE_WORK
#define AICBSP_FWLOG_EN_DEFAULT       0

#define AICBT_BTMODE_DEFAULT          AICBT_BTMODE_BT_ONLY_SW
#define AICBT_BTPORT_DEFAULT          AICBT_BTPORT_UART
#define AICBT_UART_BAUD_DEFAULT       AICBT_UART_BAUD_1_5M
#define AICBT_UART_FC_DEFAULT         AICBT_UART_FLOWCTRL_ENABLE
#define AICBT_LPM_ENABLE_DEFAULT      1
#define AICBT_TXPWR_LVL_DEFAULT      AICBT_TXPWR_LVL

#define FEATURE_SDIO_CLOCK            60000000 // 0: default, other: target c
#define FEATURE_SDIO_PHASE            2        // 0: default, 2: 180篁

struct aicbt_patch_table {
```

aic8800_fdrv

在 Makefile 中将以下两个 CONFIG 改成 y，平台选择 CONFIG_PLATFORM_ROCKCHIP



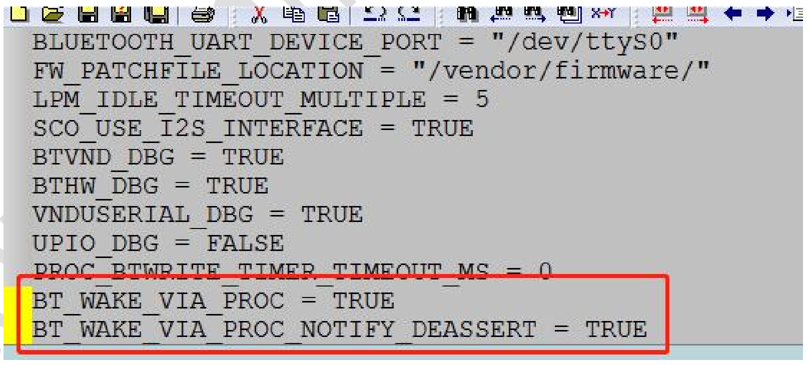
```
# Support of P2P DebugFS for enabling/disabling NoA and Op
CONFIG_RWNX_P2P_DEBUGFS := n
#
# } // WAITING FOR KCONFIG
#

# Enable A-MSDU support (need FW support)
## Select this if FW is compiled with AMSDU support
CONFIG_RWNX_SPLIT_TX_BUF ?= n
## Select this TO send AMSDU
CONFIG_RWNX_AMSDUS_TX ?= n

# Enable BFMER support (need FW support)
CONFIG_RWNX_BFMER ?= n

CONFIG_SDIO_SUPPORT =y
CONFIG_USB_SUPPORT =n
CONFIG_RX_REORDER ?=y
CONFIG_ARP_OFFLOAD =y
CONFIG_RADAR_OR_IR_DETECT =n
CONFIG_DOWNLOAD_FW =y
CONFIG_RFTEST=y
CONFIG_USB_BT =y
CONFIG_USE_5G ?= y
CONFIG_SDIO_PWRCTRL ?= y
CONFIG_CREATE_TRACE_POINTS = n
CONFIG_TXRX_THREAD_PRIO = n
# CONFIG_COEX = n for BT_ONLY, CONFIG_COEX =y for combo an
CONFIG_COEX = y
CONFIG_GPIO_WAKEUP = y
CONFIG_SET_VENDOR_EXTENSION_IE = n
CONFIG_SUPPORT_REALTIME_CHANGE_MAC = y
CONFIG_WPA3_FOR_OLD_KERNEL ?= n

```



```
endif

ifeq ($(CONFIG_RWNX_BCMC), y)
ccflags-y += -DNX_TXQ_CNT=5
else
ccflags-y += -DNX_TXQ_CNT=4
endif

# For old kernel (<=3.19)
ifeq ($(shell test $(VERSION) -lt 4 -a "$(CONFIG_VENDOR_RWNX)"
ccflags-y += -DCONFIG_VENDOR_RWNX_VHT_NO80
endif

ccflags-$(CONFIG_RX_REORDER) += -DAICWF_RX_REORDER
ccflags-$(CONFIG_ARP_OFFLOAD) += -DAICWF_ARP_OFFLOAD
ccflags-$(CONFIG_RADAR_DETECT) += -DRADAR_OR_IR_DETECT
ccflags-$(CONFIG_DOWNLOAD_FW)  += -DCONFIG_DOWNLOAD_FW

# Platform support list
CONFIG_PLATFORM_ROCKCHIP ?= y
CONFIG_PLATFORM_ALLWINNER ?= n
CONFIG_PLATFORM_AMLOGIC ?= n
CONFIG_PLATFORM_UBUNTU ?= n
CONFIG_PLATFORM_INGENIC_T20 ?= n

ccflags-y += -DAIC_TRACE_INCLUDE_PATH=$(src)

ifeq ($(CONFIG_PLATFORM_INGENIC_T20), y)
```

libbt-vendor

在 vnd_generic.txt 中将以下两个参数打开



```
BLUETOOTH_UART_DEVICE_PORT = "/dev/ttyS0"
FW_PATCHFILE_LOCATION = "/vendor/firmware/"
LPM_IDLE_TIMEOUT_MULTIPLE = 5
SCO_USE_I2S_INTERFACE = TRUE
BTVND_DBG = TRUE
BTHW_DBG = TRUE
VNDUSERIAL_DBG = TRUE
UPIO_DBG = FALSE
PROC_BTWRITE_TIMER_TIMEOUT_MS = 0
BT_WAKE_VIA_PROC = TRUE
BT_WAKE_VIA_PROC_NOTIFY_DEASSERT = TRUE
```

在 upio.c 中修改 init_rfkill()的代码

```
static int init_rfkill()
{


#if 1//For RK
    char path[64];
    char buf[16];
    int fd, sz, id;
    int rfkill_id;

    if (is_rfkill_disabled())
        return -1;

    for (id = 0; ; id++)
    {
        snprintf(path, sizeof(path), "/sys/class/rfkill/rfkill%d/type", id);
        fd = open(path, O_RDONLY);
        if (fd < 0)
        {
            ALOGE("init_rfkill : open(%s) failed: %s (%d)\n", \
                 path, strerror(errno), errno);
            return -1;
        }

        sz = read(fd, &buf, sizeof(buf));
        close(fd);

        if (sz >= 9 && memcmp(buf, "bluetooth", 9) == 0)
        {
            rfkill_id = id;
            break;
        }
    }

    asprintf(&rfkill_state_path, "/sys/class/rfkill/rfkill%d/state", rfkill_id);
    return 0;


#endif

#if 0
    char path[64];
    char buf[16];
    int fd, sz, id;
    const char *basepath = "/sys/devices/platform/aic-bt/rfkill";

    DIR *d;
    struct dirent *de;
    if (!(d = opendir(basepath)))
        goto fail;

    while ((de = readdir(d))) {
        if (strstr(de->d_name, "rfkill")) {
            snprintf(path, sizeof(path), "%s/%s/type", basepath, de->d_name);
            fd = open(path, O_RDONLY);
            if (fd < 0)
                continue;

            sz = read(fd, &buf, sizeof(buf));
            close(fd);

            if (sz >= 9 && memcmp(buf, "bluetooth", 9) == 0) {
                ALOGD("%s: rfkill path %s/%s", __func__, basepath, de->d_name);
                asprintf(&rfkill_state_path, "%s/%s/state", basepath, de->d_name);
                closedir(d);
                return 0;
            }
        }
    }
    closedir(d);

fail:
    ALOGE("%s: No rfkill control node found", __func__);
    return -1;
#endif
}
```

以上设定完后即可开启休眠唤醒的功能

Q&A

Q:成功休眠下去时，ic 功耗大约为多少
A:大约在 7-8mA

Q:发现平台休眠下去时 ic 功耗降不下来
A:确认 ic 的 bt_wake 脚是否为低，并且透过 ic 打印确认是否能敲回车

Q:平台休眠后 ping 不通
A:确认 wifi_host_wake 脚是否拉高，SDIO 的 D1 是否拉低，若以上两个电屏符合，可用镊子点一下 wifi_host_wake 是否可唤醒主控，若不可唤醒，请检查是否相关的 CONFIG 没打开