RTL8762E Deep Low Power State User Guide

Realtek Confidential

V1.0

2022/05/09

修订历史（Revision History）

日期	版本	修改	作者	Reviewer
2022/05/09	V1.0		Rui	Lory

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

目 录

修订历史（Revision History） 2
目 录 3
表目录 5
图目录 6
1 DLPS 模式概述 7
1.1 特性与限制 7
1.2 原理 7
2 DLPS 模式的进入与退出 8
2.1 DLPS 进入条件 8
2.2 DLPS 唤醒条件 8
2.3 DLPS 模式进入流程 9
2.4 DLPS 模式退出流程 10
3 硬件状态保存与恢复 11
3.1 CPU NVIC 11
3.2 PAD（AON） 11
3.3 外设 11
3.4 外挂 Sensor 11
3.5 状态保存流程 11
3.6 恢复流程 12
3.7 外设 DLPS 相关设定 13
4 PAD 唤醒功能 15
4.1 PAD 唤醒 15
4.2 Keyscan 应用 15
4.2.1 按键检测 15
4.2.2 PAD 设置 17
4.3 GPIO 应用 17
5 蓝牙相关的 DLPS 应用场景 18
5.1 Non-Link mode 18

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

5.2 Link mode 18
6 DLPS Mode API 19
6.1 lps_mode_set 19
6.2 DLPS_check_cb_reg 19
6.3 DLPS_IORegUserDlpsEnterCb 19
6.4 DLPS_IORegUserDlpsExitCb 20
6.5 System_WakeUpPinEnable 20
6.6 System_WakeUpDebounceTime 20
6.7 System_WakeUpInterruptValue 21

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

表目录

表 6-1 lps_mode_set 19
表 6-2 DLPS_check_cb_reg 19
表 6-3 DLPS_IORegUserDlpsEnterCb 19
表 6-4 DLPS_IORegUserDlpsExitCb 20
表 6-5 System_WakeUpPinEnable 20
表 6-6 System_WakeUpDebounceTime 20
表 6-7 System_WakeUpInterruptValue 21

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

图目录

图 3-1 硬件状态保存流程图 12
图 3-2 硬件设定恢复流程 12
图 4-1 Keyscan 的 debounce 机制导致中断丢失 16
图 4-2 利用 System ISR 检测 keyscan 中断 16
图 4-3 利用 PAD wakeup debounce 功能检测 keyscan 中断 17
图 4-4 利用 PAD wakeup debounce 功能检测 GPIO 中断 17

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

1 DLPS 模式概述

RTL8762E 支持三种功耗模式：Power Down 模式，DLPS（Deep Low Power State）模式和 Active 模式。此文档将详细介绍 DLPS 模式。

1.1 特性与限制

RTL8762E DLPS 模式有如下特点与限制：

1. 系统功耗 3uA@3V；
2. 系统从 DLPS 恢复需要 2ms 左右，进入 DLPS 小于 1ms；
3. PAD 和 LPC 信号可以将系统从 DLPS 状态唤醒；
4. BLE 广播和连接事件可以周期性将系统从 DLPS 状态唤醒；
5. CPU 断电会导致 SWD 断开连接， 因此在使用 keil 进行在线 debug 时，需要禁止 DLPS 模式。

1.2 原理

系统在大部分时间处于空闲状态，此时可以让系统进入 DLPS 模式以降低功耗。在 DLPS 模式下 clock，CPU，Peripherals 等模块会掉电，掉电前需要保存必要的数据用以恢复系统。当有事件需要处理时，系统会退出 DLPS 模式，并将 clock，CPU，Peripherals 等模块重新上电并恢复到进 DLPS 前的状态，然后响应唤醒事件。

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

2 DLPS 模式的进入与退出

2.1 DLPS 进入条件

只有当同时满足以下条件时，系统才会进入 DLPS 模式：

1. 系统执行在 idle task，其余 task 均处在 block 状态或 suspend 状态，且没有中断发生；
2. OS，Stack，Peripheral，APP 注册的 Check callback 执行后均返回 true；
3. SW timer 周期或者 task delay 时间大于或者等于 20ms；
4. BT 相关行为满足以下之一：
   1） Standby State
   2） BT Advertising State，Advertising Interval*0.625ms >=20ms
   3） BT Scan State，(Scan Interval – Scan Window）*0.625ms >= 15ms
   4） BT connection as Master role，Connection interval * 1.25ms > 12.5ms
   5） BT connection as Slave role，Connection interval*（1+ Slave latency）*1.25ms > 12.5ms
5. 如果蓝牙模块允许进入 DLPS 模式，但平台模块不允许，则蓝牙模块进入低功耗模式，而平台模块保持 active 状态；如果蓝牙模块不允许进入 DLPS 模式，则整个系统将仍然处于 active 状态；
6. 在 idle task 里会检查 OS，Stack，Peripherals 及 APP，如果各个模块都允许进入 DLPS，则开始保存系统状态，然后进入 DLPS。

2.2 DLPS 唤醒条件

系统可以由以下事件唤醒退出 DLPS 模式：

1. **PAD 唤醒信号**（该功能需要调用以下 API 使能）

```
1. void System_WakeUpPinEnable(uint8_t Pin_Num, uint8_t Polarity, uint8_t DebounceEn)
```

2. **BT 中断**
   1） BT 处于 advertising 状态，且 advertising anchor 到来；
   2） BT 处于 Connection 状态，且 connection anchor 到来；
   3） 有 BT 事件发生，如对端连接请求等；
3. **RTC 中断**（该功能需要调用以下 API 使能）

```
1. RTC_SystemWakeupConfig(ENABLE);
```

注意：由于 DLPS 处理 RTC 中断唤醒，可能导致中断处理不够及时，为了更精确地唤醒，可以注册 DLPS check callback，在 callback 中计算出下次唤醒的时间，由 platform 唤醒，之后处理 RTC 中断。具体做法如下：

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

1） 注释掉 RTC_WKConfig(RTC_COMP_WK_INDEX, ENABLE)，关闭 RTC 中断唤醒；

2） 计算下次唤醒时间；

```
uint32_t RTC_tick; // unit: 32.15us

bool RTC_Check_GT(uint32_t *next_wakeup_time)  //unit 31.25us
{
    uint32_t wakeup_count = RTC_GetCompValue(RTC_COMP_INDEX) - RTC_GetCounter() ;
    if (wakeup_count > 0)
    {
        *next_wakeup_time = wakeup_count * RTC_tick;
        return true;
    }
    else
    {
        return false;
    }
```

3） 注册 DLPS check callback；

```
RTC_tick = (RTC_PRESCALER_VALUE + 1) ;
    if (false == dlps_check_cb_reg(RTC_Check_GT) )
    {
        DBG_DIRECT("Error: dlps_check_cb_reg(RTC_Check_GT) failed!\n") ;
    }
```

4. **SW Timer timeout 或者 task delay 事件**
5. **LPC 中断**（该功能需要调用以下 API 使能）

```
LPC_WKCmd(ENABLE) ;
RTC_SystemWakeupConfig(ENABLE) ;
```

2.3 DLPS 模式进入流程

DLPS 模式进入流程如下：

1. 某个模块如果希望进入 DLPS 前能够询问自己，需要预先向 DLPS Framework 注册 CB。当 DLPS Framework 执行 CB 时，各模块再根据 callback 函数的结果告知 DLPS Framework 是否允许进入 DLPS。蓝牙模块是基于平台模块的，因此只有蓝牙模块允许进入 DLPS，才会继续检查平台模块。
2. 如果满足进入 DLPS 的条件，先检查蓝牙模块是否允许进入 DLPS，如果允许，则保存蓝牙状态，蓝牙进入低功耗模式；继续检查平台模块，只有平台允许进入，保存平台状态后，系统才会真正

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

进入 DLPS 状态。

1） 询问蓝牙模块是否能进入 DLPS 模式；

2） 蓝牙模块状态保存和进入 DLPS 模式；

3） 询问平台模块是否能进入 DLPS 模式；

4） 平台模块状态保存和进入 DLPS 模式；

5） 下指令进入 DLPS 模式。

2.4 DLPS 模式退出流程

系统从 DLPS 模式醒来后，首先会恢复 power 和 clock，随后 CPU 和 Peripherals 重新上电，接下来 CPU 开始执行恢复流程。只有平台模块完全退出 DLPS 后，才会检查蓝牙模块是否需要退出 DLPS。

1. **系统重启**

   系统退出 DLPS 模式后，会触发 Reset 异常进入 Reset Handler。在 Reset Handler 中会检查重启的原因。如果重新上电，系统会走重启流程。如果从 DLPS 模式唤醒，系统会走 DLPS 恢复流程。

2. **不同系统模块从 DLPS 中退出和恢复**

   1） 平台模块先退出 DLPS，并恢复状态；

   2） 判断是否蓝牙事件唤醒 DLPS。如果是，蓝牙模块退出 DLPS 并恢复状态；否则蓝牙模块继续保持低功耗状态；

3. **系统完全退出 DLPS**

   DLPS 恢复的最后阶段是在 timer task 中完成的，此时 DLPS 完全退出，会完成 CPU NVIC，peripherals 及用户自定义状态的恢复。

注意：如果系统是被 PAD 信号唤醒的，且系统使能了 System Interrupt，System ISR 会被触发。

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

3 硬件状态保存与恢复

3.1 CPU NVIC

系统进入 DLPS 后 CPU 会掉电，因此需要在进入 DLPS 前保存 NVIC 寄存器，并在退出 DLPS 后恢复 NVIC 寄存器。

3.2 PAD（AON）

PAD 在 DLPS 模式下不会掉电，因此不需要保存其状态。但是为了防止漏电，在进 DLPS 时需要对 PAD 做如下设定：

1. 系统没有使用到的 PAD，包括 package 没有出引脚的 PAD 必须设为（SW mode， Input mode，Pull Down）；
2. 系统使用到的 PAD 必须设为（SW mode，Input mode，Pull Up/Pull Down），Pull Up/Pull Down 取决于外围电路；
3. 设置了唤醒功能的 PAD 需要设定为（SW mode，Input mode，Pull Up/Pull Down），Pull Up/Pull Down 状态要与 wakeup 信号的极性相反；
4. 退出 DLPS 时要把 PAD 设置恢复成原来的状态。

3.3 外设

外设进 DLPS 时会掉电，所以进出 DLPS 时需要保存和恢复相关设定。退出 DLPS 时需要先重新使能模块并打开相应时钟，再恢复相关设定。

3.4 外挂 Sensor

外挂 Sensor 在进出 DLPS 时的处理分两种情况：

1. 如果 Sensor 不掉电，则不用恢复；
2. 如果 Sensor 掉电，则需要注册 application callback 函数，并在其中恢复 sensor 设定（重新初始化）。

3.5 状态保存流程

硬件状态保存流程如图 3-1 所示：

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

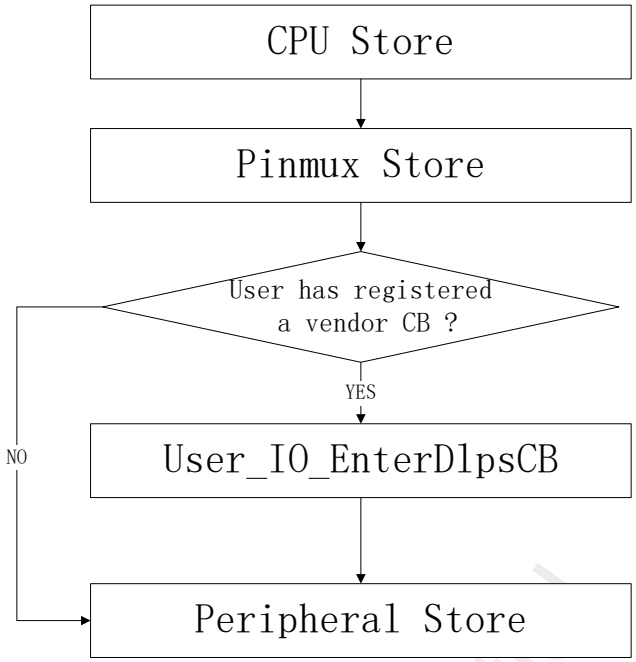

图 3-1 硬件状态保存流程图

CPU，pinmux 和 peripherals 的状态会由系统自动保存。PAD 状态如何设定取决于 APP，因此需要在 APP 中调用 DLPS_IORegUserDlpsEnterCb 注册 vendor callback function，并在该 CB 中根据需要设定 PAD 状态。如果有外挂 sensors 需要在进 DLPS 前进行保存操作，也应该放在 vendor callback 中实现。

3.6 恢复流程

硬件设定恢复流程如图 3-2 所示：

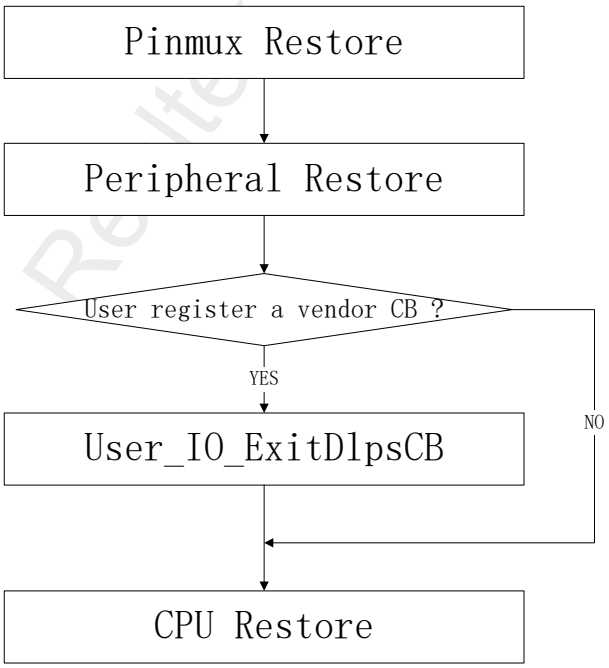

图 3-2 硬件设定恢复流程

退出 DLPS 时，系统会自动恢复 CPU，pinmux 和 peripherals。出 DLPS 时，PAD 如何设定取决于 APP，

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

因此需要在 APP 中调用 DLPS_IORegUserDlpsExitCb 注册 vendor CB，在 vendor CB 里恢复 PAD 设定。如果有外部 sensor 在出 DLPS 后需要恢复的操作，也应该放在 vendor CB 里实现。

3.7 外设 DLPS 相关设定

每个 APP project 会有一个独立的 board.h 文件，其中包含如下硬件相关的 DLPS 设定：

```
/* if use user define DLPS enter/DLPS exit callback function */
#define USE_USER_DEFINE_DLPS_EXIT_CB      1
#define USE_USER_DEFINE_DLPS_ENTER_CB     1

/* if use any peripherals below, #define it 1 */
#define USE_ADC_DLPS            0
#define USE_CTC_DLPS            0
#define USE_GPIO_DLPS           1
#define USE_I2C0_DLPS           0
#define USE_I2C1_DLPS           0
#if (ROM_WATCH_DOG_ENABLE == 1)
#define USE_TIM_DLPS            1 //must be 1 if enable watch dog
#else
#define USE_TIM_DLPS            0
#endif
#define USE_IR_DLPS             0
#define USE_KEYSCAN_DLPS        0
#define USE_QDECODER_DLPS       0
#define USE_SPI0_DLPS           0
#define USE_SPI1_DLPS           0
#define USE_SPI2W_DLPS          0
#define USE_UART0_DLPS          0
#define USE_UART1_DLPS          0
#define USE_I2S0_DLPS           0
#define USE_ENHTIM_DLPS         0
#define USE_CODEC_DLPS          0
```

如果 APP 中使用了某个外设，且需要在进出 DLPS 时，系统能自动保存、恢复其状态，则需要将该外设对应的 USE_XXX_ DLPS 宏设置为“1”。APP 还需要在 PwrMgr_Init()调用如下 API 来注册 IO DLPS CB，系统会在该 CB 里自动完成相关外设的保存和恢复：

```
DLPS_IORegister();
```

上面的例子中，使能了 ADC 与 UART 的自动保存、恢复功能。

如果需要在进出 DLPS 时做一些 APP 自定义的操作，则需要设置以下两处：

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

1. 在 board.h 中将 USE_USER_DEFINE_DLPS_EXIT_CB 与 USE_USER_DEFINE_DLPS_ENTER_CB 配置为 1；
2. 在 APP 中调用如下 API 来注册并实现 vender callback 函数。

```
1. void DlpsExitCallback(void)
2. {
3.     //do something here
4. }
5.
6. void DlpsEnterCallback(void)
7. {
8.     //do something here
9. }
10.
11. DLPS_IORegUserDlpsExitCb(DlpsExitCallback) ;
12. DLPS_IORegUserDlpsEnterCb(DlpsEnterCallback) ;
```

在上面的例子中，在进入和退出 DLPS 的过程中会分别执行 DlpsEnterCallback()和 DlpsExitCallback()，APP 可以在这两个函数中完成自定义操作，如 PAD 设置、外围设备的操作等。

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

4 PAD 唤醒功能

4.1 PAD 唤醒

1. PAD 具有 DLPS 唤醒功能，详情可参见相关硬件手册。可以调用以下 API 使能某个 PAD 的唤醒功能。当此 PAD 的电平与唤醒电平相同时，会将系统从 DLPS 状态唤醒。

```
1. void System_WakeUpPinEnable(uint8_t Pin_Num, uint8_t Polarity, uint8_t DebounceEn)
```

2. 例如希望 P3_2 为高电平时唤醒系统，可以做以下配置：

```
1. System_WakeUp_Pin_Enable(P3_2, PAD_WAKEUP_POL_HIGH, 0);
```

3. 调用 System_WakeUpPinEnable 时，将 DebounceEn 设置为 1 可以使能 wakeup debounce 功能。调用 System_WakeUpDebounceTime 可以设置 debounce time：

```
1. void System_WakeUpDebounceTime(uint8_t time)
```

4. 系统从 DLPS 醒来后，可以在 system handler 里面调用 System_WakeUpInterruptValue 来查询是哪个 PAD 唤醒了系统：

```
1. uint8_t System_WakeUpInterruptValue(uint8_t Pin_Num)
```

4.2 Keyscan 应用

4.2.1 按键检测

如图 4-1 所示，Keyscan 的 debounce 机制会导致 keyscan 中断丢失。

1. 按键按下，将系统从 DLPS 状态唤醒，开始恢复流程；
2. keyscan 模块恢复完成，开始 debounce，会在时刻 5 结束 debounce， keyscan 中断在 debounce 结束后才会产生；
3. keyscan debounce 过程；
4. 在 keyscan debounce 期间，系统没有需要处理的事件，会进入 idle task，在 keyscan debounce 结束前，再次进入 DLPS 模式；
5. Keyscan debounce 在此时结束；
6. 系统再次进入 DLPS 时，如果按键仍然被按下，系统会再次被唤醒，重复 2~5 的过程，直到按键松开，但始终无法进入 keyscan 中断。

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

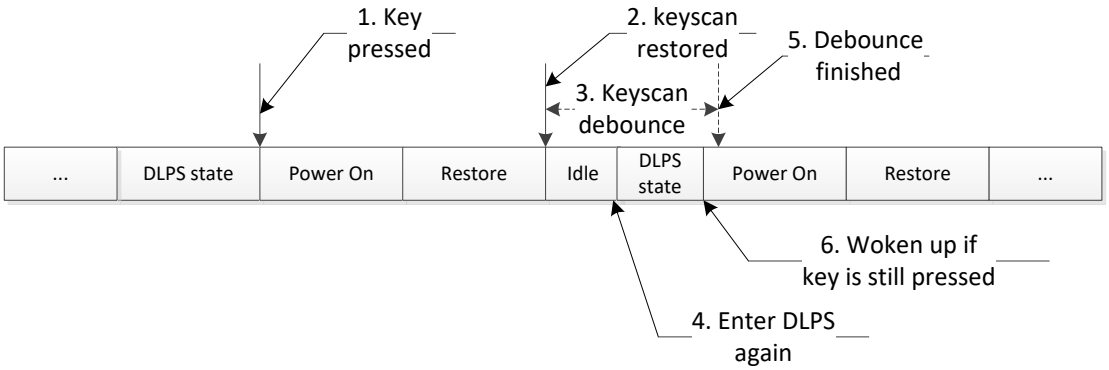

图 4-1 Keyscan 的 debounce 机制导致中断丢失

如图 4-2 所示，可以利用 System ISR 来解决该问题：

1. 按键按下，将系统从 DLPS 状态唤醒，系统开始恢复流程；
2. keyscan 模块恢复完成，开始 debounce，会在时刻 5 结束 debounce，keyscan 中断在 debounce 结束后才会产生；
3. System 中断产生，在 System ISR 中禁止系统进 DLPS；
4. Keyscan debounce 过程；
5. 系统没有事件要处理，进入 idle task，但无法进入 DLPS 模式，因为 DLPS 在 System ISR 中已被禁止；
6. Keyscan debounce 结束，keyscan 中断产生，并被系统处理。

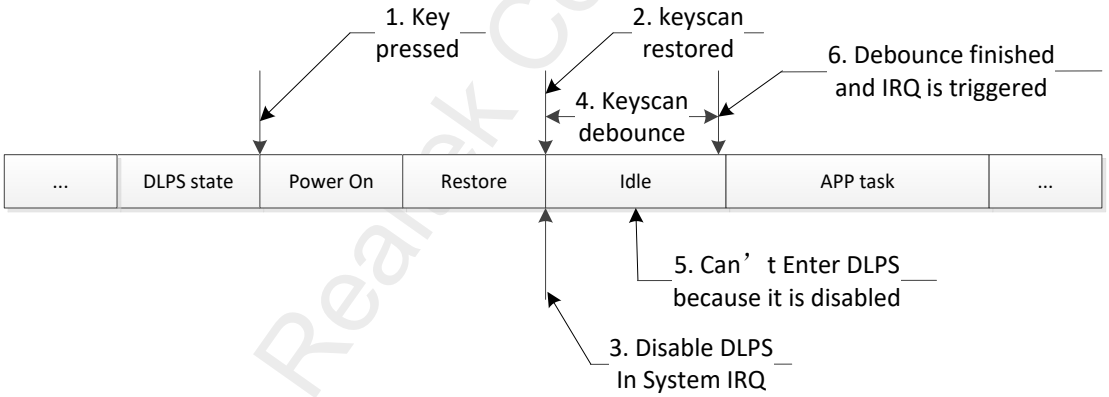

图 4-2 利用 System ISR 检测 keyscan 中断

另一种更好的策略是利用 PAD wakeup debounce 功能，如图 4-3 所示：

1. 按键按下，开始 wakeup debounce；
2. debounce 结束后，系统从 DLPS 状态被唤醒，开始恢复流程；
3. Keyscan 模块恢复完成；
4. System 中断产生，在该中断中禁止 keyscan debounce，随后即会产生 keyscan 中断；

注意：利用 wakeup debounce 可以缩短 active 的时间，从而达到降低系统功耗的效果。

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

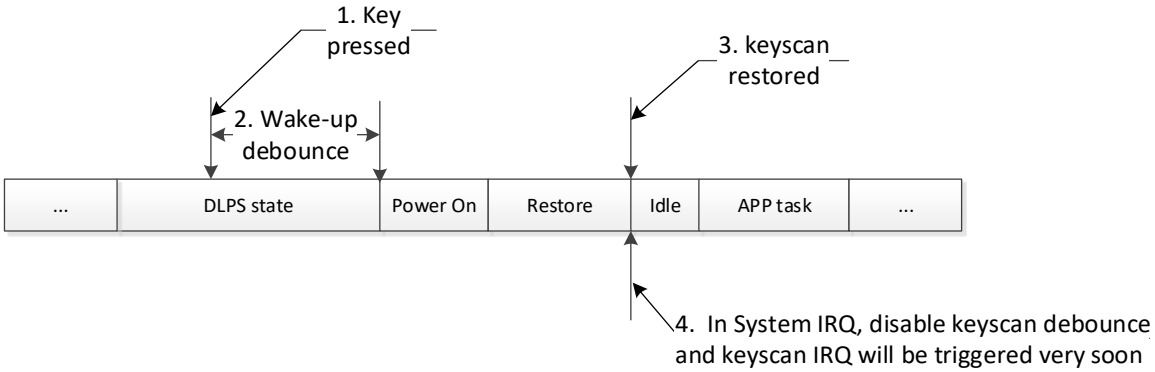

图 4-3 利用 PAD wakeup debounce 功能检测 keyscan 中断

4.2.2 PAD 设置

在进 DLPS 前，Keyscan 的列必须设置为：

```
1. PAD_Config（Px_x，PAD_SW_MODE，PAD_IS_PWRON，PAD_PULL_NONE，PAD_OUT_ENABLE，PAD_OUT_LOW）；
```

从 DLPS 醒来后， Keyscan 的列必须设置为：

```
1. PAD_Config（Px_ x，PAD_ PINMUX _MODE，PAD_IS_PWRON，PAD_PULL_NONE，PAD_OUT_ENABLE，PAD_OUT_LOW）；
```

可以在 vendor callback 中完成上述设置。Vendor callback 可以在 PwrMgr_Init() 中调用 DLPS_IORegUserDlpsEnterCb()和 DLPS_IORegUserDlpsExitCb()来注册。

4.3 GPIO 应用

当 GPIO 被配置为 edge trigger 且使能了 debounce 功能时，会存在与 keyscan 类似的问题，此时可以使用类似的机制来解决问题。如图 4-4 所示。

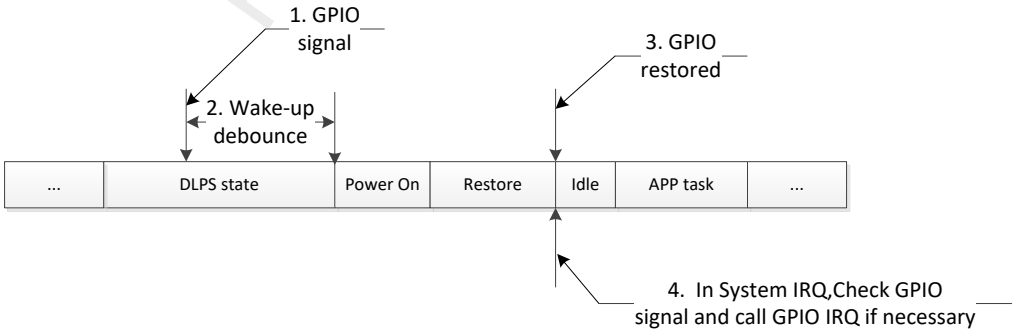

图 4-4 利用 PAD wakeup debounce 功能检测 GPIO 中断

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

5 蓝牙相关的 DLPS 应用场景

5.1 Non-Link mode

当没有连接时，BT 存在三种状态：Standby State，Advertising State 和 Scanning State。

1. 在 Standby State 时，系统不会收发数据，蓝牙不会将系统从 DLPS 状态唤醒；
2. 在 Advertising State 时，若 Adv_Interval * 0.625ms >= 20ms，允许进入 DLPS，否则不允许。若 Advertising Type 为 Direct Advertising（High duty cycle），不允许进入 DLPS；
3. 在 Scanning State 时，若（Scan Interval – Scan Window） * 0.625ms >= 15ms，允许进入 DLPS，否则不允许。

5.2 Link mode

Link mode 下有两种角色：Slave Role 和 Master Role。

1. 在 Master Role 时，若 Connection Interval * 1.25ms > 12.5ms，允许进入 DLPS；
2. 在 Slave Role 时，若 Connection Interval*（1+ Slave Latency） *1.25ms > 12.5ms，允许进入 DLPS。

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

6 DLPS Mode API

6.1 lps_mode_set

表 6-1 lps_mode_set

函数	**void lps_mode_set(PlatformPowerMode mode);**
功能	使能/禁止 DLPS 模式
参数	功耗模式，PlatformPowerMode 枚举值 1. PLATFORM_POWERDOWN：只允许 PAD 和 LPC 唤醒，此时 pad wake up debounce 需要 disable 2. PLATFORM_DLPS_PFM：DLPS 模式 3. PLATFORM_ACTIVE：系统不会进入任何低功耗模式

6.2 DLPS_check_cb_reg

表 6-2 DLPS_check_cb_reg

函数	**BOOL DLPS_check_cb_reg(DLPSEnterCheckFunc func);**
功能	向系统注册查询 callback，在系统试图进入 DLPS 前会回调该 callback，根据 callback 的返回值决定是否允许进入 DLPS 状态。有任何一个查询 callback 返回 FALSE，系统都不会进入 DLPS 状态
参数	func：进入 DLPS 前的查询 callback

6.3 DLPS_IORegUserDlpsEnterCb

表 6-3 DLPS_IORegUserDlpsEnterCb

函数	**__STATIC_INLINE void DLPS_IORegUserDlpsEnterCb(DLPS_IO_EnterDlpsCB func);**
功能	用于注册进 DLPS 时的 vendor callback; APP 自定义的 IO 保存动作需要在 vendor callback 中实现
参数	func：进入 DLPS 的 vendor callback

注意：DLPS enter callback function 不允许有较长时间的 delay 动作，否则会影响 DLPS 唤醒时间。同时也不允许调用 OS 接口，因为此时已关闭 OS 调度和中断。

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

6.4 DLPS_IORegUserDlpsExitCb

表 6-4 DLPS_IORegUserDlpsExitCb

函数	**__STATIC_INLINE void DLPS_IORegUserDlpsExitCb(DLPS_IO_ExitDlpsCB func);**
功能	用于注册出 DLPS 时的 vendor callback，APP 自定义的 IO 恢复动作需要在 vendor callback 中实现
参数	DLPS_IO_ExitDlpsCB func：退出 DLPS 的 vendor callback

6.5 System_WakeUpPinEnable

表 6-5 System_WakeUpPinEnable

函数	**void System_WakeUpPinEnable(uint8_t Pin_Num, uint8_t Polarity, uint8_t DebounceEn);**
功能	用于配置 PAD 的 wakeup 功能
参数	1. Pin_Num：详情参见 rtl876x.h 中“Pin_Number”部分 2. Polarity：唤醒极性 1) PAD_WAKEUP_POL_HIGH：高电平唤醒 2) PAD_WAKEUP_POL_LOW：低电平唤醒 3. DebounceEn：使能/禁用 wakeup debounce

6.6 System_WakeUpDebounceTime

表 6-6 System_WakeUpDebounceTime

函数	**void System_WakeUpDebounceTime(uint8_t time）；**
功能	用于设置 debounce 时间
参数	time：debounce 时间，单位：ms。

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.

6.7 System_WakeUpInterruptValue

表 6-7 System_WakeUpInterruptValue

函数	**uint8_t System_WakeUpInterruptValue(uint8_t Pin_Num);**
功能	用于查询某个 PAD 是否是唤醒系统的 PAD，若返回 true，则该 PAD 即为唤醒系统的 PAD。
参数	Pin_Num：待查询的 PAD

Realtek Confidential

Copyright Realtek Semiconductor Corporation.

All Rights Reserved.