

## **PAN1070 LowPower 例程说明文档**

PAN-CLT-VER-A0, Rev 0.1

PANCHIP

PanchipMicroelectronics

[www.panchip.com](http://www.panchip.com)

## 修订历史

| 版本   | 修订日期       | 描述     |
|------|------------|--------|
| V0.1 | 2024-10-17 | 初始版本创建 |

PANCHIP

## 目录

|   |    |
|---|----|
| 第 1 章 例程环境准备 .....                        | 4  |
| 1.1 硬件环境 .....                            | 4  |
| 1.2 软件环境 .....                            | 4  |
| 1.2.1 上位机软件工具 .....                       | 4  |
| 1.2.2 例程代码 .....                          | 4  |
| 1.2.3 测试程序编译烧录 .....                      | 5  |
| 第 2 章 功能演示 .....                          | 7  |
| 2.1 Sleep 模式睡眠与唤醒 .....                   | 7  |
| 2.1.1 Sleep 模式下通过 GPIO 唤醒 .....           | 7  |
| 2.1.2 Sleep 模式下通过 UART 唤醒 .....           | 8  |
| 2.1.3 Sleep 模式下通过 TIMER 唤醒 .....          | 10 |
| 2.1.4 Sleep 模式下通过 SLPTMR 唤醒 .....         | 12 |
| 2.2 DeepSleep 模式睡眠与唤醒 .....               | 13 |
| 2.2.1 DeepSleep 模式下通过 GPIO 唤醒 .....       | 13 |
| 2.2.2 DeepSleep 模式下通过 TIMER 唤醒 .....      | 14 |
| 2.2.3 DeepSleep 模式下通过 SLPTMR 唤醒 .....     | 16 |
| 2.3 Standby Mode1 模式睡眠与唤醒 .....           | 17 |
| 2.3.1 Standby Mode1 模式下通过 GPIO 唤醒 .....   | 17 |
| 2.3.2 Standby Mode1 模式下通过 SLPTMR 唤醒 ..... | 18 |
| 2.4 Standby Mode0 模式睡眠与唤醒 .....           | 19 |
| 2.4.1 Standby Mode0 模式下通过特殊 IO 唤醒 .....   | 20 |
| 第 3 章 开发者说明 .....                         | 22 |

# 第1章 例程环境准备

## 1.1 硬件环境

- 1、JLink 仿真器（用于程序烧录与调试）
- 2、电流计（本文使用可视化电流测量设备 PPK2 [Nordic Power Profiler Kit II]进行演示）
- 3、USB 转串口模块（用于测试 Sleep 模式下的 UART1 Rx 中断唤醒）
- 4、PAN1070 QFN32 EVB 核心板与底板（V1.0）各 1 块：
  - a) 将 EVB 核心板插到 EVB 底板排座上
  - b) 使用 USB-TypeC 线，一端连接 PC，另一端连接 EVB 底板的 USB->UART 插座 U5
  - c) **确认 EVB 底板上只有如下 4 个跳线被接上**（多余的跳线可能导致功耗测量不准确！）：
    - i. Voltage 电源排针 P11 第 3 排 VCC 接至 3V3
    - ii. Voltage 电源排针 P11 第 4 排 VDD 接至 3V3
    - iii. 左上角功能排针 P2 倒数第一排 TX 接至 P16（例程交互串口 UART0 Tx）
    - iv. 左上角功能排针 P2 倒数第二排 RX 接至 P17（例程交互串口 UART0 Rx）
  - d) 确认 EVB 底板上的 POWER 拨动开关 **SW2 拨至 BAT 档位**，并确保 EVB 底板背部的 Battery 电池仓内**没有纽扣电池**（此时核心板上的 PAN107x 芯片处于未供电的状态）
  - e) 将电流计 PPK2 的：
    - i. VOUT 连接至 EVB 底板的 VBAT 排针
    - ii. GND 连接至 EVB 底板的 GND 排针

## 1.2 软件环境

### 1.2.1 上位机软件工具

- 1、串口调试助手（UartAssist）或终端工具（SecureCRT），波特率 921600（用于例程测试交互）
- 2、nRF Connect Desktop - Power Profiler（用于配合 PPK2 电流计测量芯片功耗）

### 1.2.2 例程代码

测试工程文件：

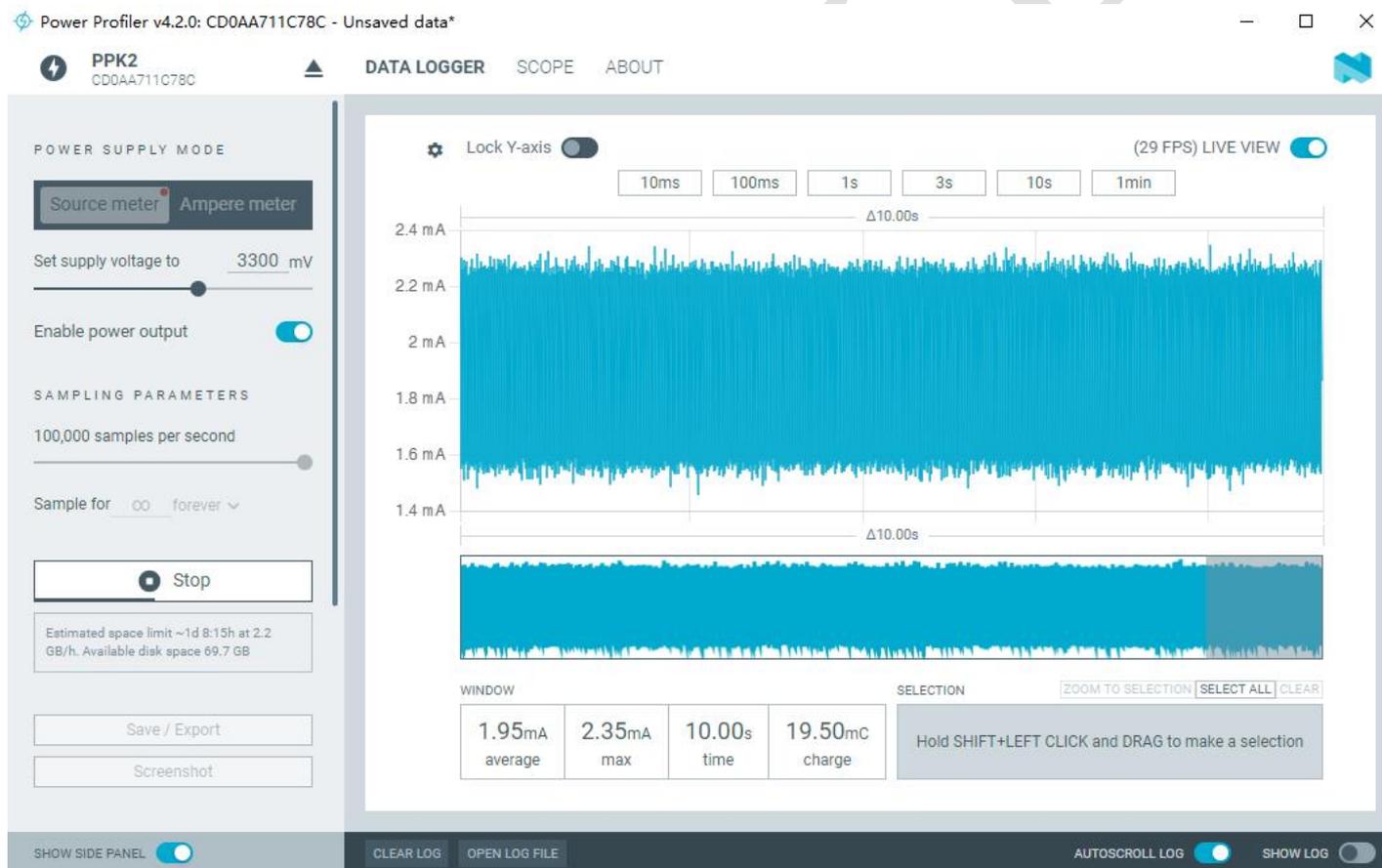
<PAN1070-DK>\03\_MCU\mcu\_samples\LowPower\Keil\LowPower.uvprojx

测试源文件目录：

<PAN1070-DK>\03\_MCU\mcu\_samples\LowPower\src

### 1.2.3 测试程序编译烧录

1. 打开 Keil 工程，点击 Build 按钮，确认程序可以成功编译。
2. 将 JLink 连接至 EVB 底板（SWD\_CLK: P00，SWD\_DAT: P01，SWD\_GND: GND），点击 Keil Download 按钮，将程序烧录至芯片，成功后断开 JLink 与芯片的连接，防止后续功耗测量不准确。
3. 打开 nRF Connect Desktop 软件的 Power Profiler 功能界面，在 Source Meter 选项卡下，将供电电压修改为 3300mV，然后点击 Enable power output 以开启供电，此时点击 Start 按钮进行采样，可以看到芯片当前工作状态下的功耗约为 2mA：



4. 观察串口是否正常打印出如下的例程演示主菜单（在进行后续测试之前，最好先按一下 EVB 核心板或底板上的复位按钮，可以从串口 Log 中看到复位源为 nRESET Pin Reset）：



## 第2章 功能演示

### 2.1 Sleep 模式睡眠与唤醒

在主菜单下，输入 ‘1’ 命令进入 Sleep 模式演示子菜单：

```
1
+-----+
| Press key to choose a sleep wakeup source: |
|                                             |
| Input 'A'   wake up by GPIO interrupt.    |
| Input 'B'   wake up by UART interrupt.    |
| Input 'C'   wake up by TIMER interrupt.   |
| Input 'D'   wake up by SLPTMR interrupt.  |
| Press ESC key to back to the top level case list. |
+-----+
```

#### 2.1.1 Sleep 模式下通过 GPIO 唤醒

1. 在 Sleep 模式演示子菜单下，输入 ‘A’ 命令，使芯片进入 Sleep 模式；
2. 稍等片刻，再按下 EVB 底板上的 KEY1 按键；
3. 观察 UART Log，可以看到成功触发 GPIO 中断将芯片唤醒：

```
a
Configure GPIO P06 as input mode and enable falling edge interrupt detection..
P0_6 INT occurred.
```

```
+-----+
| Press key to choose a sleep wakeup source: |
|                                             |
| Input 'A'   wake up by GPIO interrupt.    |
| Input 'B'   wake up by UART interrupt.    |
| Input 'C'   wake up by TIMER interrupt.   |
| Input 'D'   wake up by SLPTMR interrupt.  |
| Press ESC key to back to the top level case list. |
+-----+
```

4. 观察电流计波形，可以看到芯片平均电流先是下降到 1mA 左右，说明芯片进入了 Sleep 模式，然后再被 GPIO 中断唤醒，平均电流恢复到 2mA 左右：



### 2.1.2 Sleep 模式下通过 UART 唤醒

1. 将一个额外的 USB 转串口模块，USB 端连接 PC，串口端的 Tx 引脚连接至 EVB 底板的 P07 (UART1 Rx) 排针，串口端的 GND 引脚连接至 EVB 底板的 GND 排针；
2. 在 Sleep 模式演示子菜单下，输入 ‘B’ 命令，使芯片进入 Sleep 模式；
3. 在 PC 上重新打开一个串口调试工具，连接步骤 1 中新增的 USB 转串口模块，波特率设置为 115200，切换至 HEX 十六进制发送模式，然后随意向串口发送一些数据：



4. 观察 UART Log, 可以看到成功触发 UART 中断将芯片唤醒, 并打印接收到的串口数据:

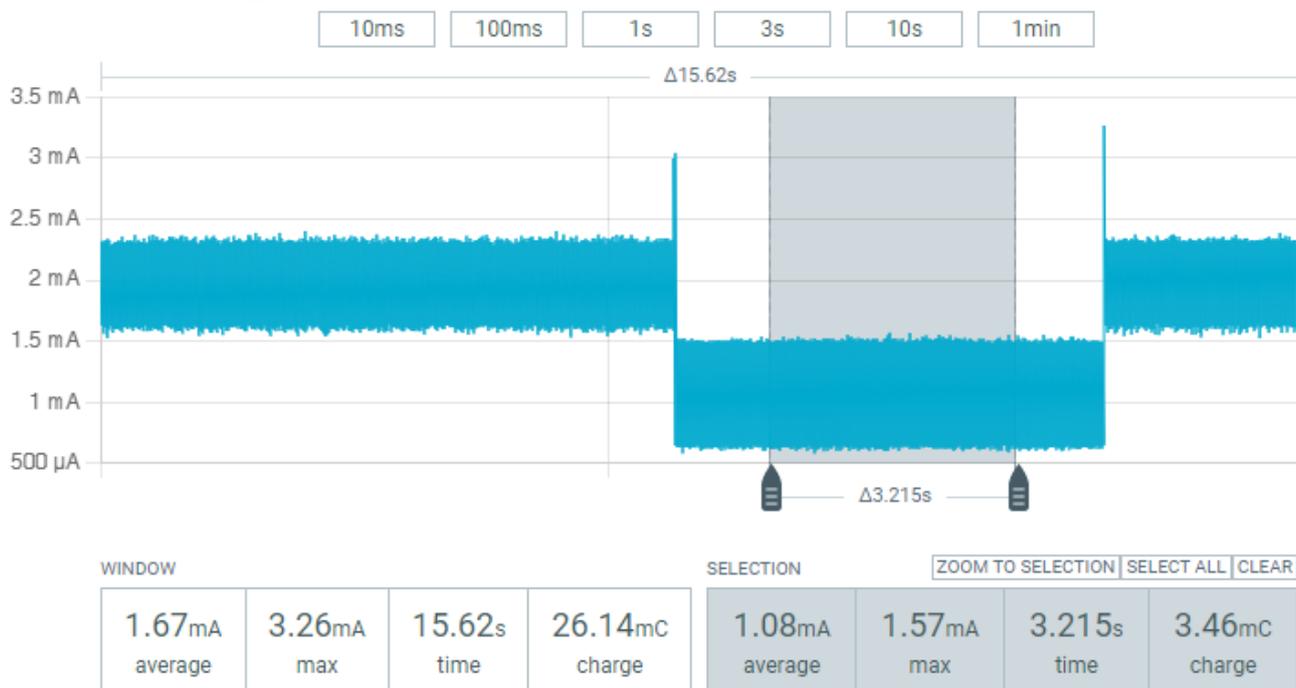
```

b
Configure UART1 receive data with interrupt..
UART1 INT Rx: 0x12.
UART1 INT Rx: 0xa5.
    
```

```

+-----+
| Press key to choose a sleep wakeup source: |
|                                             |
| Input 'A'   wake up by GPIO interrupt.   |
| Input 'B'   wake up by UART interrupt.   |
| Input 'C'   wake up by TIMER interrupt.  |
| Input 'D'   wake up by SLPTMR interrupt. |
| Press ESC key to back to the top level case list. |
+-----+
    
```

5. 观察电流计波形, 可以看到芯片平均电流先是下降到 1mA 左右, 说明芯片进入了 Sleep 模式, 然后再被 UART1 Rx 中断唤醒, 平均电流恢复到 2mA 左右:



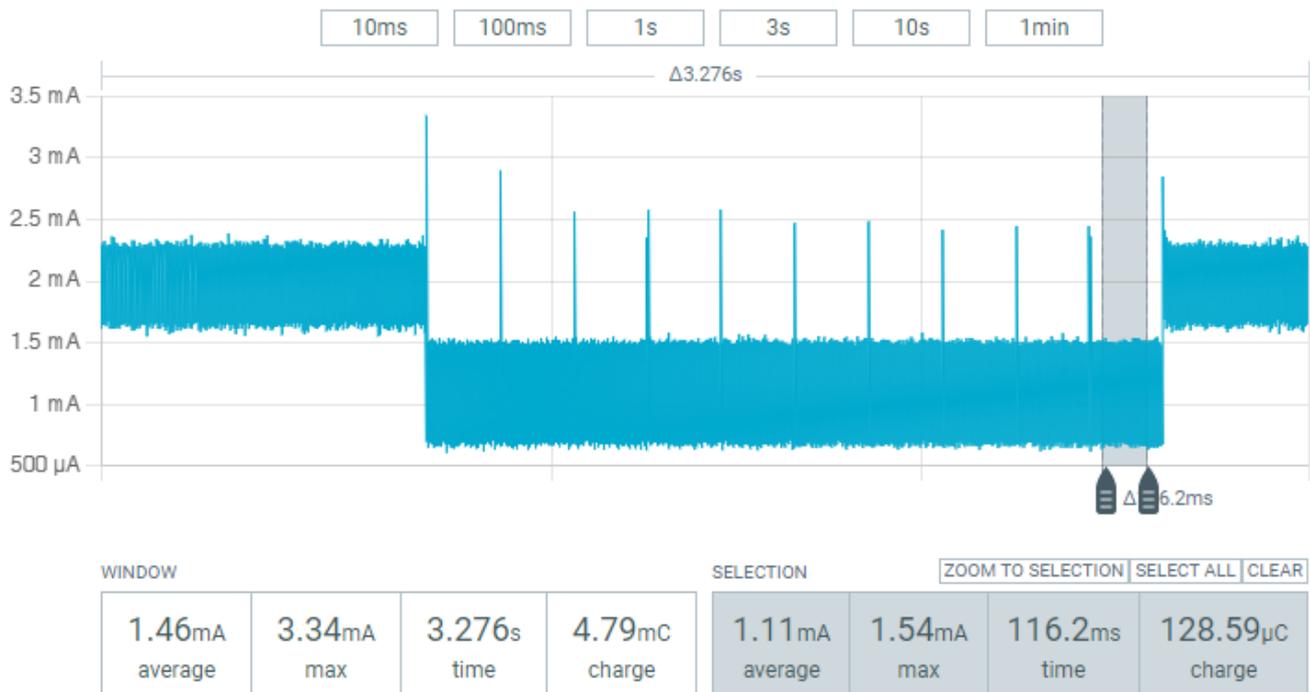
### 2.1.3 Sleep 模式下通过 TIMER 唤醒

1. 在 Sleep 模式演示子菜单下，输入 ‘C’ 命令，使芯片进入 Sleep 模式；
2. 稍等片刻，Timer 超时时间（程序配置为 200ms）到来后将芯片唤醒，随后会立刻重新进入 Sleep 模式，反复 10 次后芯片退出演示流程；
3. 观察 UART Log，可以看到成功触发 10 次 Timer 中断将芯片唤醒 10 次；

```
C
Configure TIMER0 with timeout interrupt..
Enter sleep mode, cnt = 0
TIMER0 INT occurred.
Wakeup, cnt = 0
Enter sleep mode, cnt = 1
TIMER0 INT occurred.
Wakeup, cnt = 1
Enter sleep mode, cnt = 2
TIMER0 INT occurred.
Wakeup, cnt = 2
Enter sleep mode, cnt = 3
TIMER0 INT occurred.
Wakeup, cnt = 3
Enter sleep mode, cnt = 4
TIMER0 INT occurred.
Wakeup, cnt = 4
Enter sleep mode, cnt = 5
TIMER0 INT occurred.
Wakeup, cnt = 5
Enter sleep mode, cnt = 6
TIMER0 INT occurred.
Wakeup, cnt = 6
Enter sleep mode, cnt = 7
TIMER0 INT occurred.
Wakeup, cnt = 7
Enter sleep mode, cnt = 8
TIMER0 INT occurred.
Wakeup, cnt = 8
Enter sleep mode, cnt = 9
TIMER0 INT occurred.
Wakeup, cnt = 9
```

```
+-----+
| Press key to choose a sleep wakeup source: |
| Input 'A'   wake up by GPIO interrupt.   |
| Input 'B'   wake up by UART interrupt.   |
| Input 'C'   wake up by TIMER interrupt.  |
| Input 'D'   wake up by SLPTMR interrupt. |
| Press ESC key to back to the top level case list. |
+-----+
```

4. 观察电流计波形，可以看到芯片平均电流先是下降到 1mA 左右，说明芯片进入了 Sleep 模式，然后再被 Timer 中断唤醒，平均电流恢复到 2mA 左右，如此反复变化 10 次：



### 2.1.4 Sleep 模式下通过 SLPTMR 唤醒

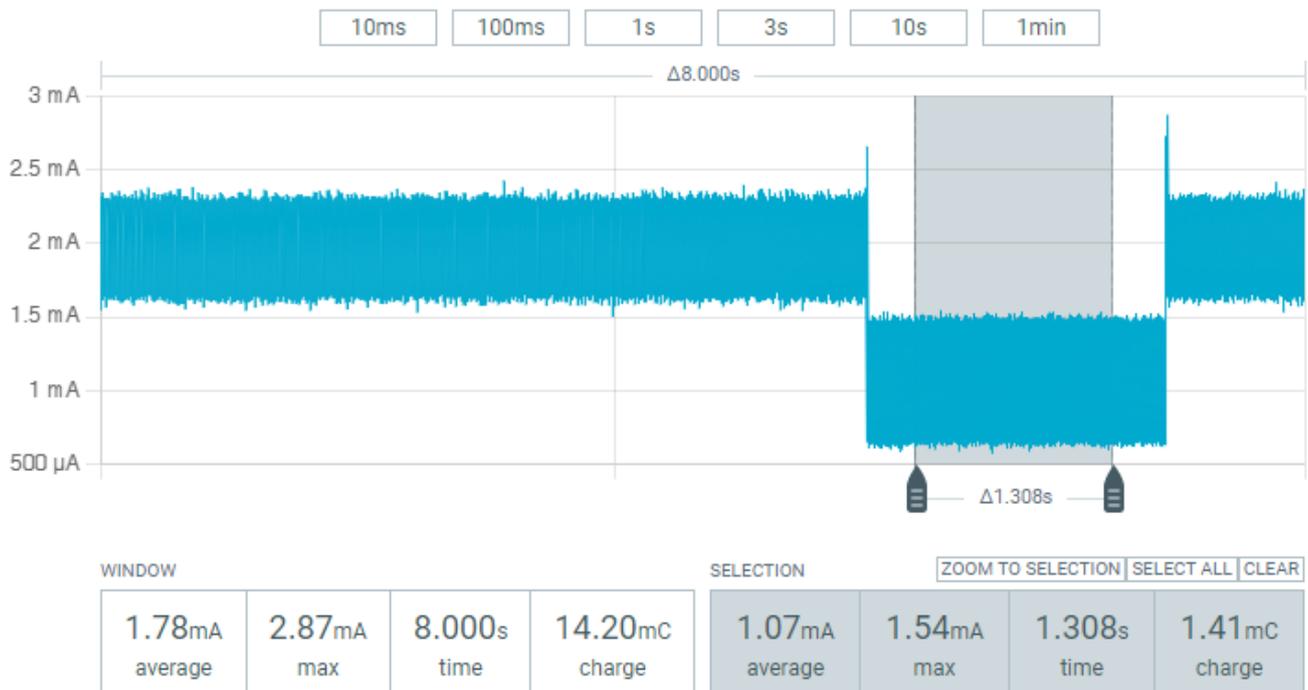
1. 在 Sleep 模式演示子菜单下，输入 ‘D’ 命令，使芯片进入 Sleep 模式；
2. 稍等片刻，SleepTimer 超时时间（程序配置为 2s）到来后将芯片唤醒；
3. 观察 UART Log，可以看到成功触发 SleepTimer 0 中断将芯片唤醒：

```
d
Configure sleepTimer 0 with timeout interrupt..
SleepTimer0 INT occurred.
```

```
+-----+
| Press key to choose a sleep wakeup source: |
+-----+
```

```
Input 'A'   wake up by GPIO interrupt.
Input 'B'   wake up by UART interrupt.
Input 'C'   wake up by TIMER interrupt.
Input 'D'   wake up by SLPTMR interrupt.
Press ESC key to back to the top level case list.
```

4. 观察电流计波形，可以看到芯片平均电流先是下降到 1mA 左右，说明芯片进入了 Sleep 模式，然后再被 SleepTimer 0 超时中断唤醒，平均电流恢复到 2mA 左右：



## 2.2 DeepSleep 模式睡眠与唤醒

在主菜单下，输入 ‘2’ 命令进入 DeepSleep 模式演示子菜单：

```
2
+-----+
| Press key to choose a deepsleep wakeup source: |
| Input 'A'   wake up by GPIO.                   |
| Input 'B'   wake up by TIMER.                  |
| Input 'C'   wake up by SLPTMR.                 |
| Press ESC key to back to the top level case list. |
+-----+
```

### 2.2.1 DeepSleep 模式下通过 GPIO 唤醒

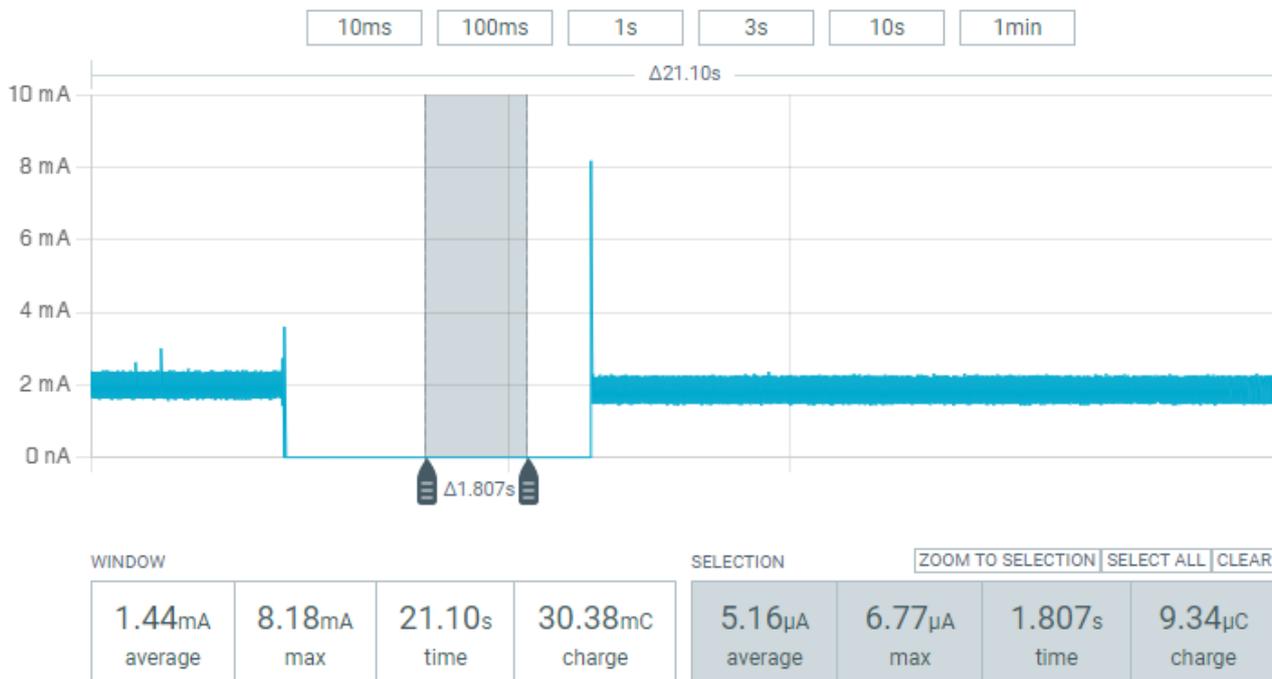
1. 在 DeepSleep 模式演示子菜单下，输入 ‘A’ 命令，使芯片进入 DeepSleep 模式；
2. 稍等片刻，再按下 EVB 底板上的 KEY1 按键；
3. 观察 UART Log，可以看到成功触发 GPIO 中断将芯片唤醒：

```
a
Configure GPIO P06 as input mode and enable falling edge interrupt detection..
P0_6 INT occurred.
```

```
+-----+
| Press key to choose a deepsleep wakeup source: |
| Input 'A'   wake up by GPIO.                   |
| Input 'B'   wake up by TIMER.                  |
| Input 'C'   wake up by SLPTMR.                 |
| Press ESC key to back to the top level case list. |
+-----+
```

4. 观察电流计波形，可以看到芯片平均电流先是下降到 5uA 左右，说明芯片进入了

DeepSleep 模式，然后再被 GPIO 中断唤醒，平均电流恢复到 2mA 左右：



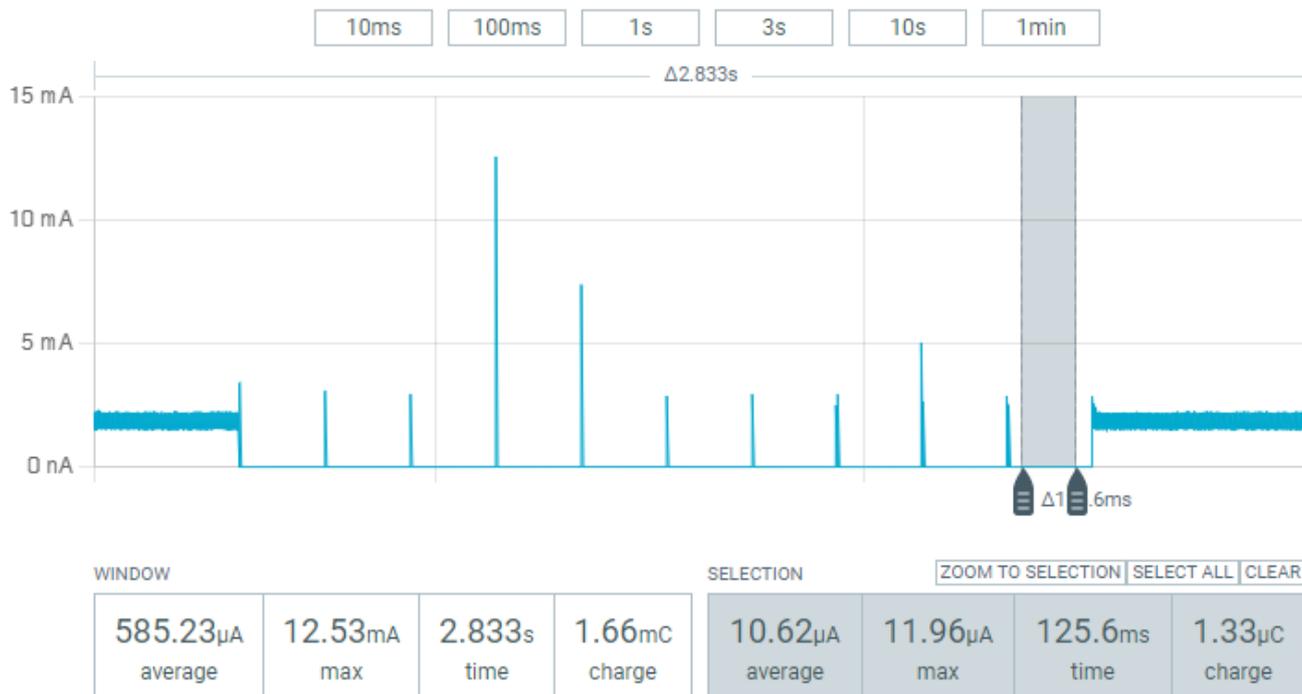
### 2.2.2 DeepSleep 模式下通过 TIMER 唤醒

1. 在 DeepSleep 模式演示子菜单下，输入 ‘B’ 命令，使芯片进入 DeepSleep 模式；
2. 稍等片刻，Timer 超时时间（程序配置为 200ms）到来后将芯片唤醒，随后会立刻重新进入 DeepSleep 模式，反复 10 次后芯片退出演示流程；
3. 观察 UART Log，可以看到成功触发 10 次 Timer 中断将芯片唤醒 10 次；

```
b
Configure TIMER0 with timeout interupt..
Enter deepsleep mode, cnt = 0
TIMER0 INT occurred.
waked up.
Enter deepsleep mode, cnt = 1
TIMER0 INT occurred.
waked up.
Enter deepsleep mode, cnt = 2
TIMER0 INT occurred.
waked up.
Enter deepsleep mode, cnt = 3
TIMER0 INT occurred.
waked up.
Enter deepsleep mode, cnt = 4
TIMER0 INT occurred.
waked up.
Enter deepsleep mode, cnt = 5
TIMER0 INT occurred.
waked up.
Enter deepsleep mode, cnt = 6
TIMER0 INT occurred.
waked up.
Enter deepsleep mode, cnt = 7
TIMER0 INT occurred.
waked up.
Enter deepsleep mode, cnt = 8
TIMER0 INT occurred.
waked up.
Enter deepsleep mode, cnt = 9
TIMER0 INT occurred.
waked up.
```

```
+-----+
|       |
| Press key to choose a deepsleep wakeup source: |
|       |
| Input 'A'   wake up by GPIO. |
| Input 'B'   wake up by TIMER. |
| Input 'C'   wake up by SLPTMR. |
| Press ESC key to back to the top level case list. |
|       |
+-----+
```

4. 观察电流计波形，可以看到芯片平均电流先是下降到 10uA 左右，说明芯片进入了 DeepSleep 模式，然后再被 Timer 中断唤醒，平均电流恢复到 2mA 左右，如此反复变化 10 次：



注：此处低功耗底电流为 10uA，比前面 GPIO 唤醒模式的 5uA 低功耗底电流要大一些，原因是 Timer 唤醒需要进入芯片的 DeepSleep Mode 2，而此模式下芯片 DVDD 电压会高一些，因此功耗也会大一些。

### 2.2.3 DeepSleep 模式下通过 SLPTMR 唤醒

1. 在 DeepSleep 模式演示子菜单下，输入 ‘C’ 命令，使芯片进入 DeepSleep 模式；
2. 稍等片刻，SleepTimer 超时时间（程序配置为 500ms）到来后将芯片唤醒；
3. 观察 UART Log，可以看到成功触发 SleepTimer 1 中断将芯片唤醒：

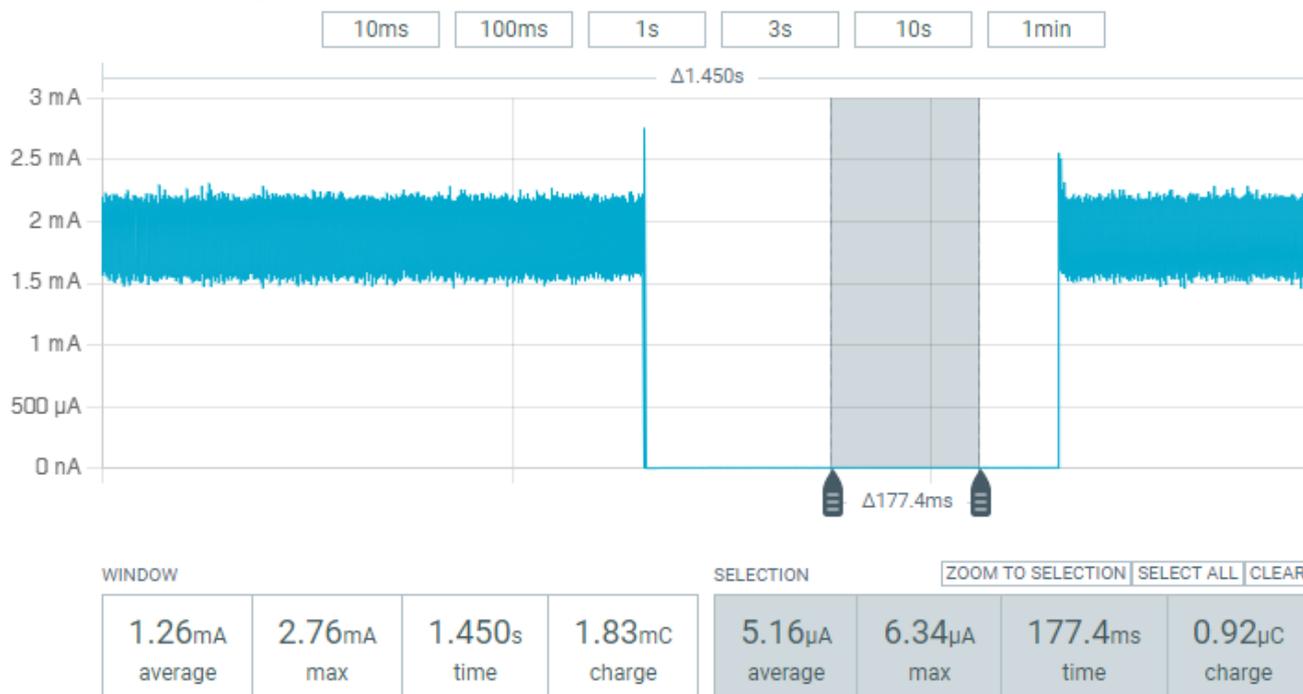
```

C
Configure sleepTimer 1 with timeout interrupt..
SleepTimer1 INT occurred.
    
```

```

+-----+
| Press key to choose a deepsleep wakeup source: |
| Input 'A'   wake up by GPIO. |
| Input 'B'   wake up by TIMER. |
| Input 'C'   wake up by SLPTMR. |
| Press ESC key to back to the top level case list. |
+-----+
    
```

4. 观察电流计波形，可以看到芯片平均电流先是下降到 5uA 左右，说明芯片进入了 DeepSleep 模式，然后再被 SleepTimer 1 超时中断唤醒，平均电流恢复到 2mA 左右：



## 2.3 Standby Mode1 模式睡眠与唤醒

在主菜单下，输入 ‘3’ 命令进入 Standby Mode 1 模式演示子菜单：

```

3
+-----+
| Press key to choose a standby mode 1 wakeup source: |
| Input 'A'   wake up by GPIO. |
| Input 'B'   wake up by SLPTMR. |
| Press ESC key to back to the top level case list. |
+-----+
    
```

### 2.3.1 Standby Mode1 模式下通过 GPIO 唤醒

1. 在 Standby Mode 1 模式演示子菜单下，输入 ‘A’ 命令，使芯片进入 Standby Mode 1；
2. 稍等片刻，再按下 EVB 底板上的 KEY1 按键；
3. 观察 UART Log，可以看到成功触发芯片唤醒并复位，并且从复位 Log 中可以看到当前芯片的复位原因为 Standby Mode 1 GPIO 唤醒，唤醒 IO 为 P06；





```
4
+-----+
| Press key to choose a standby mode 0 wakeup source: |
| Input 'A'   wake up by special IO P02.             |
| Press ESC key to back to the top level case list.  |
+-----+
```

### 2.4.1 Standby Mode0 模式下通过特殊 IO 唤醒

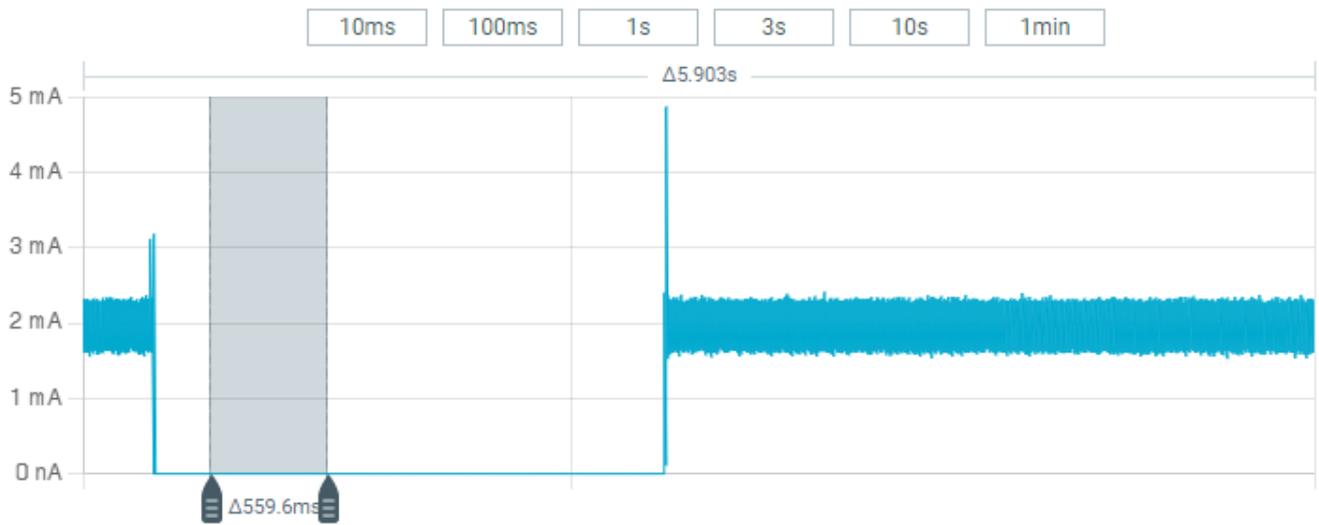
1. 在 Standby Mode 0 模式演示子菜单下，输入 ‘A’ 命令，使芯片进入 Standby Mode 0；
2. 稍等片刻，再按下 EVB 底板上的 WKUP（P02）按键；
3. 观察 UART Log，可以看到成功触发芯片唤醒并复位，并且从复位 Log 中可以看到当前芯片的复位原因为 Standby Mode 0 Special IO 唤醒：

```
a
Configure special IO P02 as input mode and enable low-level wakeup detection..
Try to load HW calibration data.. DONE.
- Chip Info       : 0x1
- Chip CP Version : 255
- Chip FT Version : 6
- Chip MAC Address : E11000001005
- Chip UID        : 3D0001465454455354
- Chip Flash UID  : 4250315A3538380B00698D4356039878
- Chip Flash Size : 512 KB
CPU @ 48000000HZ
```

Reset Reason: Standby Mode 0 Special IO (P00/P01/P02) wakeup.

```
+-----+
| LowPower Sample Code. |
+-----+
| Press specific key to start test.. |
| Input '1'   Enter and wakeup from SoC sleep Mode. |
| Input '2'   Enter and wakeup from SoC DeepSleep Mode. |
| Input '3'   Enter and wakeup from SoC Standby Mode 1. |
| Input '4'   Enter and wakeup from SoC Standby Mode 0. |
+-----+
```

4. 观察电流计波形，可以看到芯片平均电流先是下降到 300nA 左右，说明芯片进入了 Standby Mode 0，然后再被 GPIO P02 的特殊模式唤醒并复位，平均电流恢复到 2mA 左右：



WINDOW

|                   |               |                |                  |
|-------------------|---------------|----------------|------------------|
| 1.17mA<br>average | 4.86mA<br>max | 5.903s<br>time | 6.93mC<br>charge |
|-------------------|---------------|----------------|------------------|

SELECTION

ZOOM TO SELECTION | SELECT ALL | CLEAR

|                   |               |                 |                  |
|-------------------|---------------|-----------------|------------------|
| 0.29μA<br>average | 1.59μA<br>max | 559.6ms<br>time | 0.16μC<br>charge |
|-------------------|---------------|-----------------|------------------|

PANCHIP

## 第3章 开发者说明

1. PAN107x 芯片内部支持 LDO 模式与 DCDC 模式供电, 芯片上电默认使用 LDO 模式; 在 SDK 中可以通过使能 CONFIG\_SOC\_DCDC\_PAN1070 配置开关的方式, 将芯片切换至 DCDC 模式, 此模式下芯片动态功耗表现会更加优秀;
2. 芯片 Sleep 模式下, 仅仅将 CPU 时钟停止, 所有外设的电源和时钟均处于开启状态, 因此任何中断均可以将芯片从 Sleep 状态下唤醒;
3. 芯片 DeepSleep 模式下, 芯片内部所有高速时钟均停止工作, 所有数字模块的供电电压均从正常工作的 1.2v 下降到 0.4v~0.7v, 此模式下只有特定的模块可以将芯片唤醒;
4. Standby Mode 1 模式下, 芯片内部所有高速时钟均停止工作, 除 GPIO 和特殊保电区域外的所有数字模块均掉电, 此模式下只有 GPIO 和 SleepTimer 可以将芯片唤醒;
5. Standby Mode 0 模式下, 芯片内部所有高速时钟均停止工作, 默认低速时钟也会被禁止 (特定需求下可手动打开), 几乎内部所有模块均掉电, 此模式下只有特定的 3 个 GPIO (P00、P01、P02) 可以将芯片唤醒, 且只支持电平唤醒, 不支持边沿唤醒;
6. 若在开发过程中, 出现低功耗底电流偏大的情况, 则应首先怀疑是有 IO 漏电情况, 请检查各个 IO 是否在使用过程中被配置成数字输入状态, 但进入低功耗之前没有关闭该 IO 的数字输入功能, 且出现了 IO 引脚电平浮空的情况;
7. 若在开发过程中, 出现无法正常进入低功耗模式的情况, 则应首先怀疑是否在进入低功耗流程之前, 系统还有中断源没有处理, 若有则芯片不会进入低功耗, 程序也不会停在 WFI, 而是继续往下跑;
8. 若在开发过程中, 出现无法正常唤醒的情况, 则可能原因有:
  - a) 各个模块的中断源和唤醒源没有正确配置, 需要检查代码;
  - b) 若发现 Timer 或 WDT 无法正常从 DeepSleep 状态下唤醒, 其他唤醒源 (如 GPIO) 唤醒正常, 则可能是 LPLDOH 电压偏低导致;
  - c) 若 DeepSleep 唤醒后发现芯片复位, 则也可能是 LPLDOH 电压偏低;
  - d) 以上两种情况均可以通过增加 CONFIG\_SOC\_INCREASE\_LPLDOH\_CALIB\_CODE 的值解决;