



SF32LB52x

低功耗场景测试报告

latest (2026-06-04)

RPT5202

目录

1. SF32LB52x 芯片简介	4
1.1. 芯片架构概述	5
1.2. 测试内容概览	5
1.3. 重要提醒	6
1.3.1. 影响功耗的关键因素	7
2. 测试环境配置指南	7
2.1. 相关硬件	8
2.2. 相关软件	8
2.3. 功耗测量方法	8
2.3.1. 更换外接天线方法	9
3. BLE 低功耗蓝牙测试	12
3.1. 测试说明	13
3.2. 例程编译与烧写	15
3.2.1. 编译	15
3.2.2. 烧写镜像	16
3.2.3. 改变发射功率	17
3.3. 设备初始化	17
3.4. ADV场景	17
3.5. 连接场景	19
3.6. 测量结果汇总	21
3.6.1. BLE 功耗表	21
4. 经典蓝牙功耗测试	21
4.1. 测试说明	22
4.2. 例程编译与烧录	24
4.2.1. 编译	24
4.2.2. 烧写镜像	25
4.2.3. 改变发射功率	26
4.3. 设备初始化	26
4.4. Scan	26
4.5. Sniff Mode	29

4.6. 测量结果汇总	33
4.6.1. BT 功耗	33
5. 处理器性能与功耗测试	33
5.1. 测试目标	34
5.2. 测试说明	34
5.3. 例程编译与烧录	36
5.3.1. 编译	36
5.3.2. 烧录镜像	37
5.4. CoreMark	37
5.4.1. HPSYS	37
5.5. While Loop	38
5.5.1. HPSYS	39
5.6. 关机	40
5.7. 测量结果汇总	40
5.7.1. 处理器功耗	41

SF32LB52x 低功耗测试报告

欢迎使用 SF32LB52x 系列芯片低功耗测试报告！

芯片简介

关于芯片的介绍

测试环境

了解测试所需的硬件环境、软件配置和测试工具

BLE 低功耗测试

蓝牙低功耗场景下的详细测试方法和结果

BT 经典蓝牙测试

经典蓝牙应用场景的功耗测试数据

Coremark 性能测试

基准性能测试中的功耗表现分析

1. SF32LB52x 芯片简介

1.1. 芯片架构概述

SF32LB52x 是一颗基于 Arm Cortex-M33 STAR-MC1 处理器的创新性大小核架构 SoC 芯片。

高性能大核

专门设计用于高性能运算场景：

- 图形处理
- 音频处理
- 神经网络运算

低功耗小核 (LCPU)

优化用于低功耗场景：

- 蓝牙协议栈 Controller 层
- 实时任务处理
- 待机状态管理

1.2. 测试内容概览

本文档提供了 SF32LB52x 系列芯片在多种使用场景下的功耗测试方法与参考结果：

测试场景总览

类别	测试场景	说明
无线通信	BLE 低功耗蓝牙	广播、连接、数据传输等典型场景

类别	测试场景	说明
无线通信	BT 经典蓝牙	传统蓝牙协议的各种应用模式
性能测试	Coremark 基准测试	标准化 CPU 性能测试中的功耗表现
基础测试	While Loop 循环	基础循环操作的功耗基线
待机模式	关机模式	最低功耗状态下的功耗数据

1.3. 重要提醒



测试结果仅供参考

本文档的测试结果与软硬件环境紧密相关，实际应用中可能存在差异。

1.3.1. 影响功耗的关键因素

芯片因素

- 芯片自身的个体差异
- 制造工艺的批次变化

硬件因素

- 外围元器件的选型与质量
- 芯片内置 Buck 的外接电感
- PCB 设计与布线

测试环境

- 测试设备的准确度与校准
- 电源的稳定性与纹波
- 环境温湿度条件

软件配置

- 固件版本与编译选项
- I/O 引脚的配置状态
- 时钟与电源管理设置

环境条件

- 工作温度范围
- 电磁干扰水平
- 供电电压波动

2. 测试环境配置指南

准确的功耗测试需要标准化的环境配置。本章节介绍了测试所需的各项环境要求。

2.1. 相关硬件

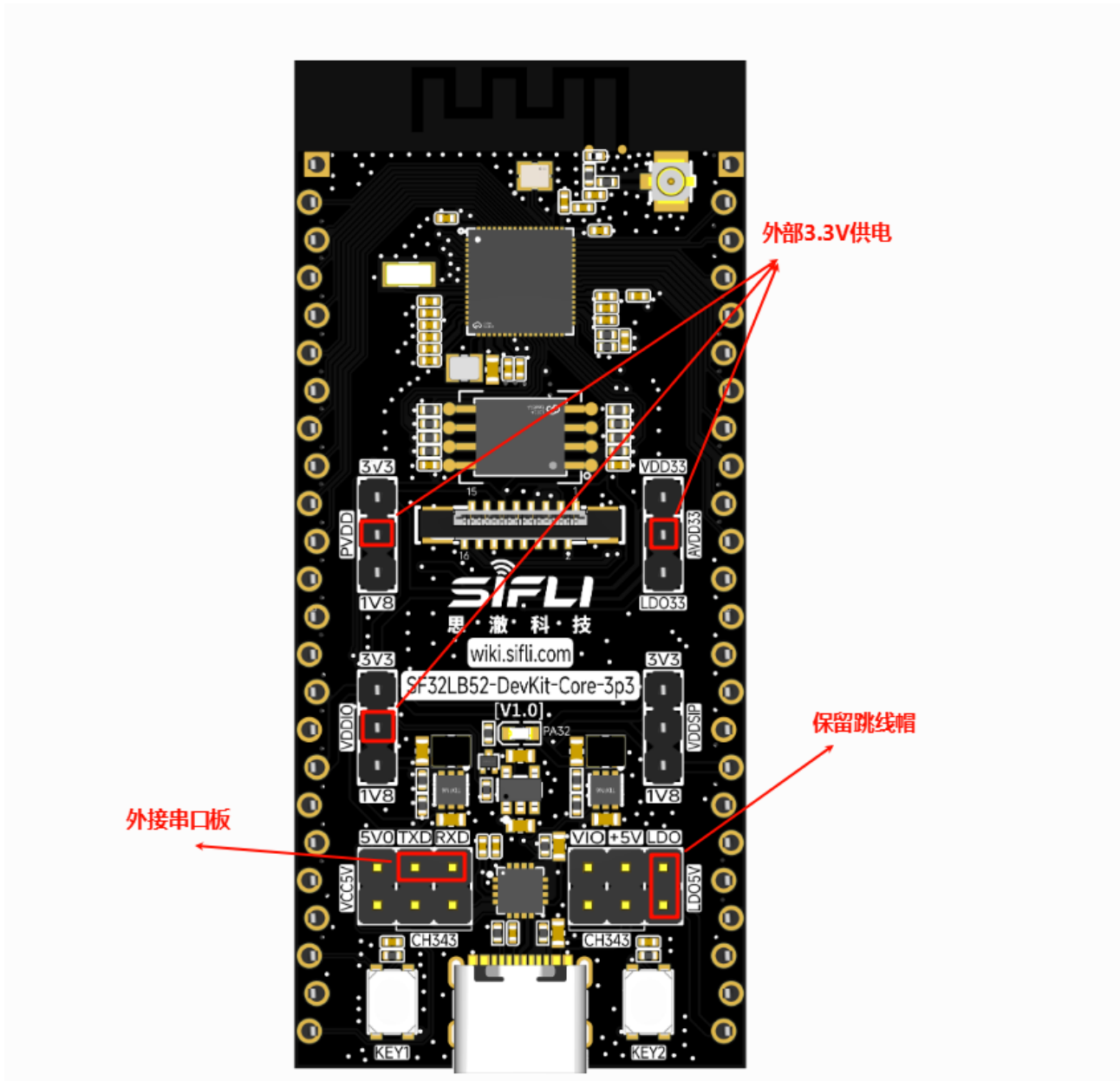
- SF32LB52-DevKit-Core-3p3(板载芯片型号 SF32LB52BU56/SF32LB52JUD6)，使用方法参考[SF32LB52-DevKit-Core-3p3开发板使用指南](#)
- Agilent 66319D直流电源
- Keysight 34465A 数字电流计
- 串口板
- 安卓手机，安装 LightBlue App（com.punchthrough.lightblueexplorer_LightBlue_1.9.3.apk，可从 GooglePlay 应用商店下载）或者安装 BLE调试助手（从安卓自带的应用商店即可下载）。

2.2. 相关软件

- BLE 和 BT 功耗测试例程：example\pm\bt\src
- CoreMark 功耗测试例程：example\pm\coremark\src

2.3. 功耗测量方法

使用功耗测试仪器同时对PVDD，AVDD33,VDDIO进行3.3V供电，供电的针脚如图框出，其余跳线帽全部去除，保留LDO5V的跳线帽，TXD与RXD外接串口板用来输入命令



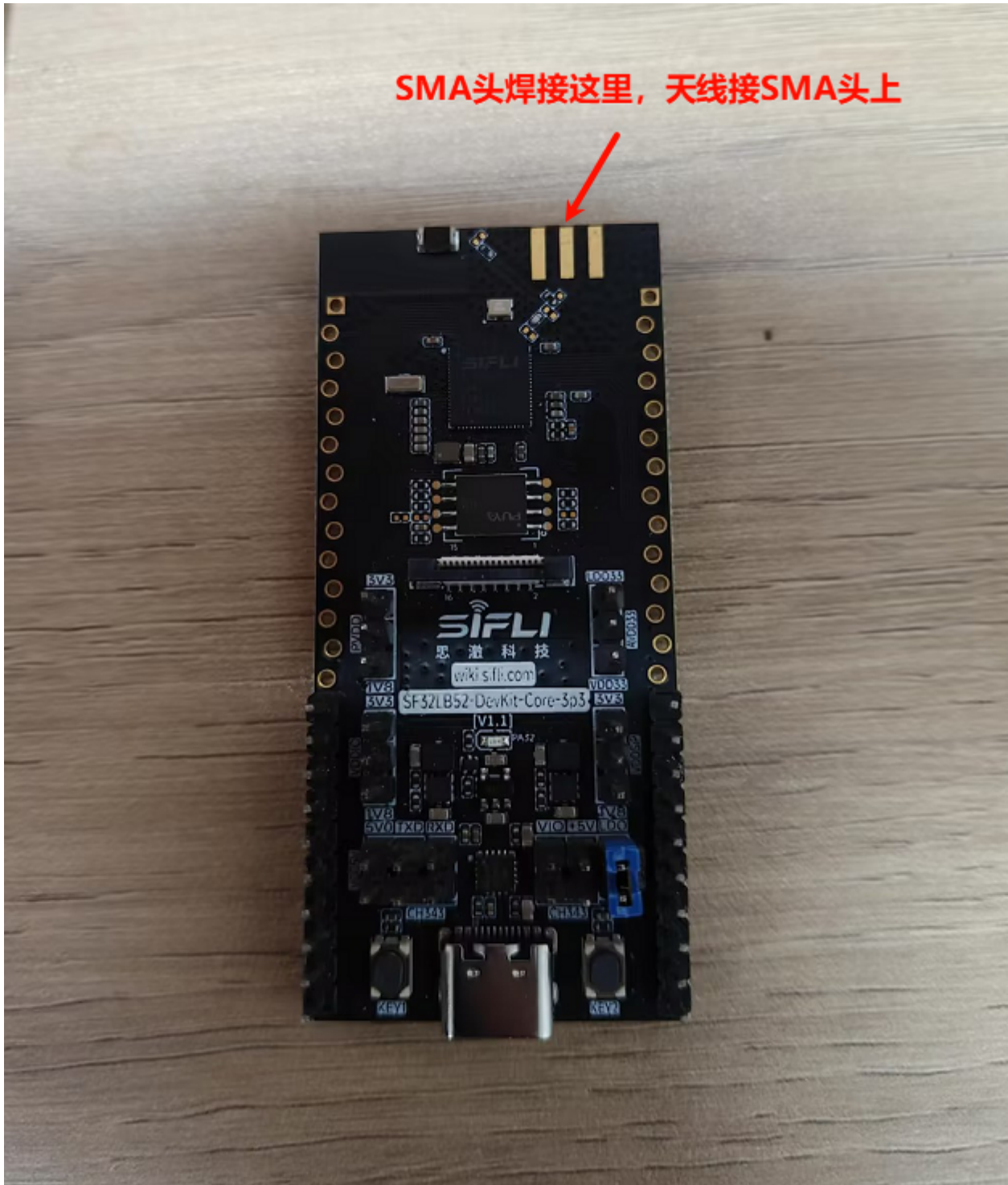
开发板上默认使用板载天线进行射频信号收发，也可自行选择更换为外接天线。

2.3.1. 更换外接天线方法

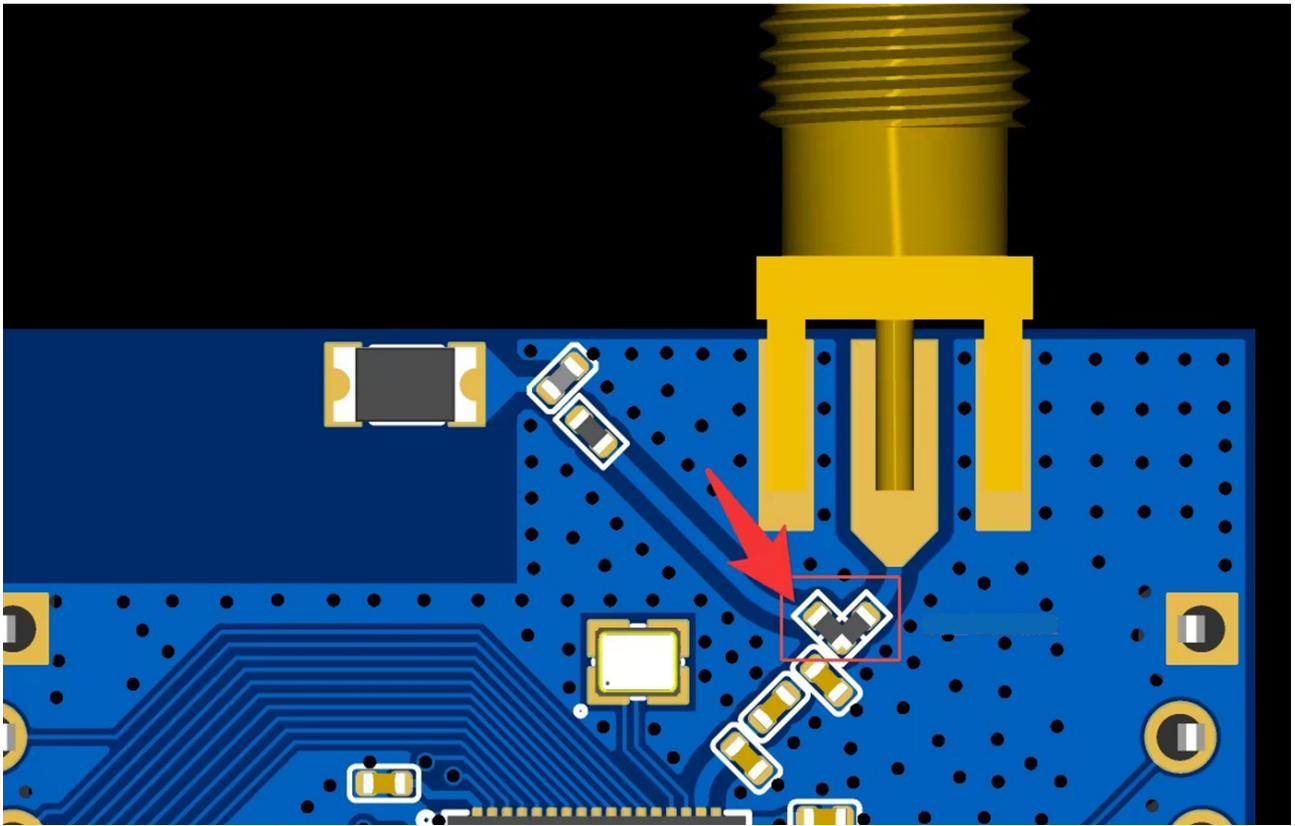
1. 需要的硬件：SMA头+天线（或50欧姆负载），规格为：50欧姆的射频负载或是50欧姆的RF天线。



1. 先将天线以及SMA头按照如下图所示进行焊接



1. 随后将如图挑选电阻方向更换为指向外接天线，默认是指向板载天线的



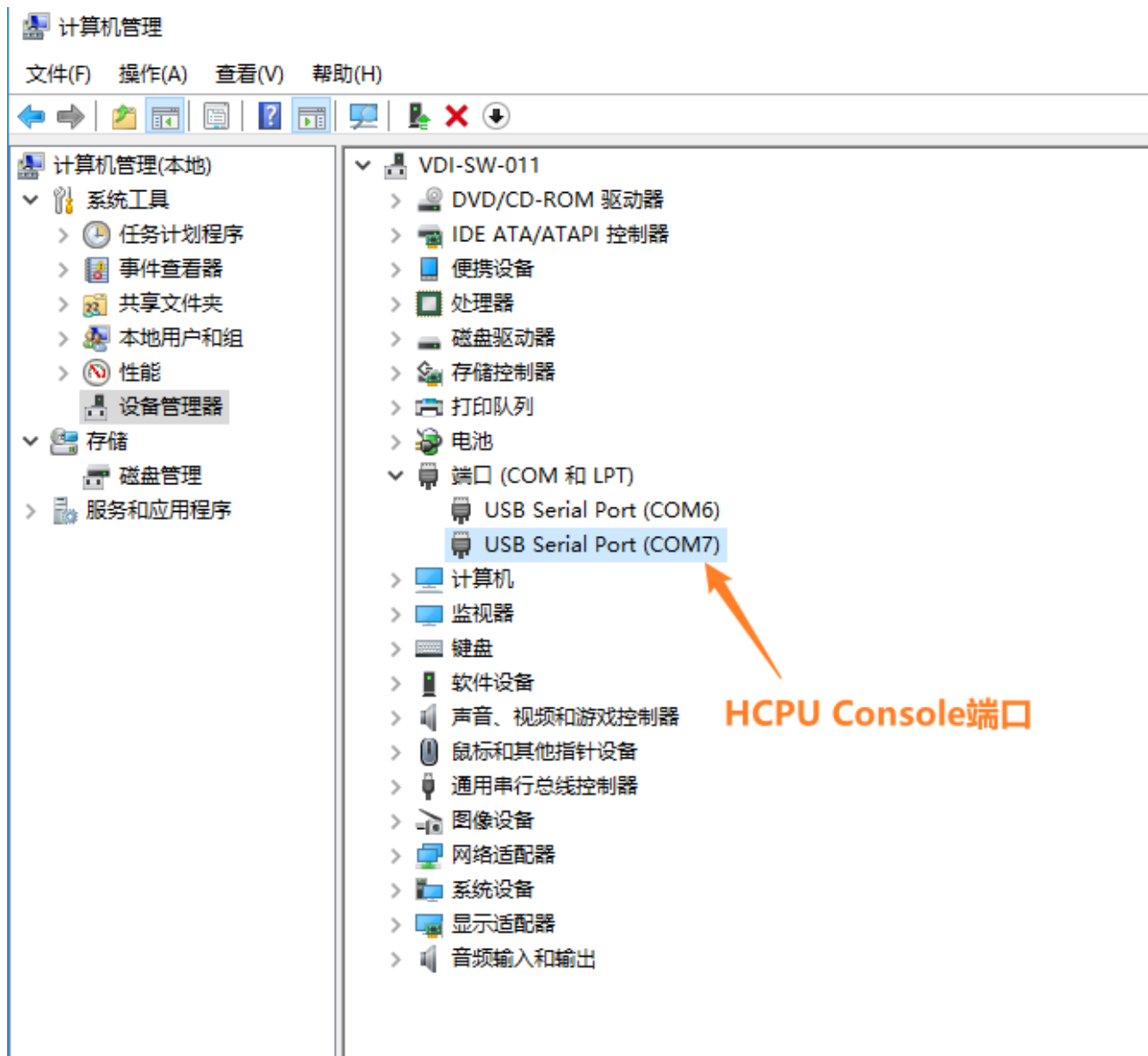
3. BLE 低功耗蓝牙测试

BLE (Bluetooth Low Energy) 是 SF32LB52x 的核心功能之一。本章节详细介绍各种 BLE 应用场景下的功耗测试方法和结果。

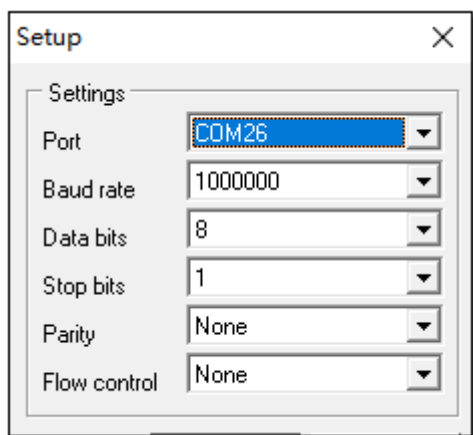
3.1. 测试说明

使用 BLE 例程可以测试典型的 ADV 和连接场景功耗，系统上电后例程自动开启 ADV 发送，使用手机的 LightBlue 软件发现并连接设备，通过 HCPU 的 console 发送命令修改 ADV 发送间隔和连接周期，发送的命令都要以回车换行符结尾。

PC 与调试板使用 USB Type-C 线连接后会枚举出两个串口，其中 HCPU 使用第二个串口作为 console 端口，如下图所示。



串口设置参见下图：波特率均设置为 1000000。



为便于控制测试条件，使用 PA24 引脚作为 HCPU 的唤醒 PIN。当唤醒 PIN 为低电平时 HCPU 无法进入低功耗模式，此时可以通过 console 给 HCPU 发送命令修改参数，当 PA24 悬空或者接高电平（即 3.3V 电压，下文

如未特别说明，高电平均指 3.3V 电压）时，HCPU 进入低功耗模式，LCPU 则周期性的进出低功耗模式，此时 console 无法使用。

测试例程 LCPU 的主频为 24MHz，HCPU 的低功耗模式为 Deepsleep，LCPU 的低功耗模式为 Standby。

HCPU 支持如下命令修改 BLE ADV 和连接周期，同时还可使用 `btskey` 命令操作菜单修改 BT 的配置。

- `ble_config adv interval_in_ms`: 其中 `interval_in_ms` 是毫秒为单位的 ADV 间隔； 例如：发送命令 `ble_config adv 100`，可将 ADV 周期改为 100ms
- `ble_config conn interval_in_ms`: 其中 `interval_in_ms` 是毫秒为单位的连接周期； 例如：在连接之后发送 `ble_config conn 100`，可将连接周期改为 100ms

3.2. 例程编译与烧写

3.2.1. 编译

如果使用已编译好的 `image` 文件，可以直接跳到烧录部分进行烧录开始测试。

进入 `example\pm\bt\project\hcpu` 目录，执行

```
scons --board=sf32lb52-core_n16r16 -j8
```

编译生成 HCPU 的 `image` 文件(不需要单独编译 LCPU 的工程，编译 HCPU 时会自动编译 LCPU 并打包 LCPU 的 `bin` 到 HCPU 的镜像中)，编译生成的 `image` 文件保存在 `build` 目录下。

... example > pm > bt > project > hcpu > build_sf32lb52-core_n16r16_hcpu >				
   排序 ▾  查看 ▾ ...				
名称	修改日期	类型	大小	
board	2025/9/22 18:04	文件夹		
bootloader	2025/9/22 18:04	文件夹		
ftab	2025/9/22 18:05	文件夹		
lcpu	2025/9/22 18:05	文件夹		
main	2025/9/22 18:04	文件夹		
sifli_sdk	2025/7/21 9:44	文件夹		
 .config	2025/9/22 18:04	Configuration 源文件	23 KB	
 .config.old	2025/9/22 17:52	OLD 文件	23 KB	
 compile_commands.json	2025/9/22 18:04	JSON 源文件	2,697 KB	
 cppdefines.txt	2025/9/22 18:04	文本文档	1 KB	
 custom_mem_map.h	2025/9/22 18:04	C Header 源文件	3 KB	
 download.bat	2025/9/22 18:05	Windows 批处理文件	1 KB	
 download.jlink	2025/9/22 18:05	JLINK 文件	1 KB	
 ImgBurn.log	2025/9/22 18:07	文本文档	5 KB	
 ImgBurnList.ini	2025/9/22 18:05	配置设置	1 KB	
 Kconfig	2025/9/22 18:04	文件	1 KB	
 kconfiglist	2025/9/22 18:04	文件	10 KB	
 link.sct	2025/9/22 18:04	Windows Script Co...	6 KB	
 main.asm	2025/9/22 18:05	ASM 文件	18,264 KB	
 main.axf	2025/9/22 18:05	AXF 文件	8,373 KB	

3.2.2. 烧写镜像

在命令行编译的目录下执行

```
build_sf32lb52-core_n16r16_hcpu\uart_download.bat
```

烧写 build 目录下编译生成的镜像文件。

3.2.3. 改变发射功率

工程默认配置的发射功率是0dbm，可以使用 `ble_tx_pwr_save x` 命令修改发射功率，x是需要修改的发射功率大小，例如改变发射功率为10dbm：`ble_tx_pwr_save 10`。改完后会自动重启生效。

3.3. 设备初始化

第一次使用开发板测试蓝牙需要在上电后执行以下初始化命令（烧写文件系统镜像后蓝牙地址会丢失，也需要重新配置），命令在 HCPU 的 console 中执行，唤醒 PIN 需接低电平避免 HPSYS 进入低功耗模式，否则 HCPU 无法处理命令

- `nvds reset_all 1`
- `nvds update addr 6 <蓝牙地址>`

例如 `nvds update addr 6 1234567890C8`，注意蓝牙地址分类需要保持一定的格式，建议 C8 不要改动，其他的用户可以自行改变，命令执行完成后按 **Reset** 键重启设备

3.4. ADV场景

- 注意：测试ADV场景时最好在屏蔽箱或者屏蔽房进行测试，否则有其他设备干扰会导致测试结果偏高

1. ◦ 打开串口调试工具，连接 HCPU 的 console 串口，连接测量设备与被测模块
2. ◦ 唤醒 PIN 接低电平，按底板的 **Reset** 键复位，启动成功后出现如图下图的 log

```

\ \ \
- SiFli Corporation
/ / \ build Jan 21 2021, v0.6.2
2020 - 2021 Copyright by SiFli team
Patch hook install..
[1326] D/DS: data service subscribe (BLE_NV), service (0)
[32m[1355] I/nvds: Wait nvds service.
[0m[1379] D/DS: data service subscribe (ANCS), service (2010b6ac)
[1411] D/DS: dataac subscribe dev open
[32m[1464] I/ble_cm: read_bond_infor_from_flash: 7
[0m[32m[1502] I/sibles: enable BLE Core..
[0m[32m[1525] I/ble_app: status 0
[0m[32m[1548] I/nvds: nvds msg_id 3841
[0m[32m[1571] I/nvds: get_msg 0
[0m[32m[1590] I/nvds: Subscribed nvds service ret 0.
[0m[32m[1624] I/ancs_srv: ancs event handler 116

[0m[32m[1650] I/ble_cm: BLE_NVDS_AYSNC_FLUSH_TAG_CNF
[0m[32m[1678] I/ancs_srv: ancs event handler 112

[0m[32m[1704] I/ble_cm: BLE_NVDS_ASYNC_READ_CNF
[0m[31m[1728] E/ble_cm: len check error! 210, 0
[0m[32m[1753] I/ble_cm: BLE_NVDS_ASYNC_READ_CNF 0x10
[0m[32m[1780] I/ble_cm: BLE_NVDS_ASYNC_READ_CNF 0x1
[0m[32m[1806] I/ble_cm: BLE_NVDS_ASYNC_READ_CNF 0x17
[0m[32m[19753] I/sibles:
BLE ready!

[0m[32m[19777] I/ancs_srv: ancs event handler 0

```

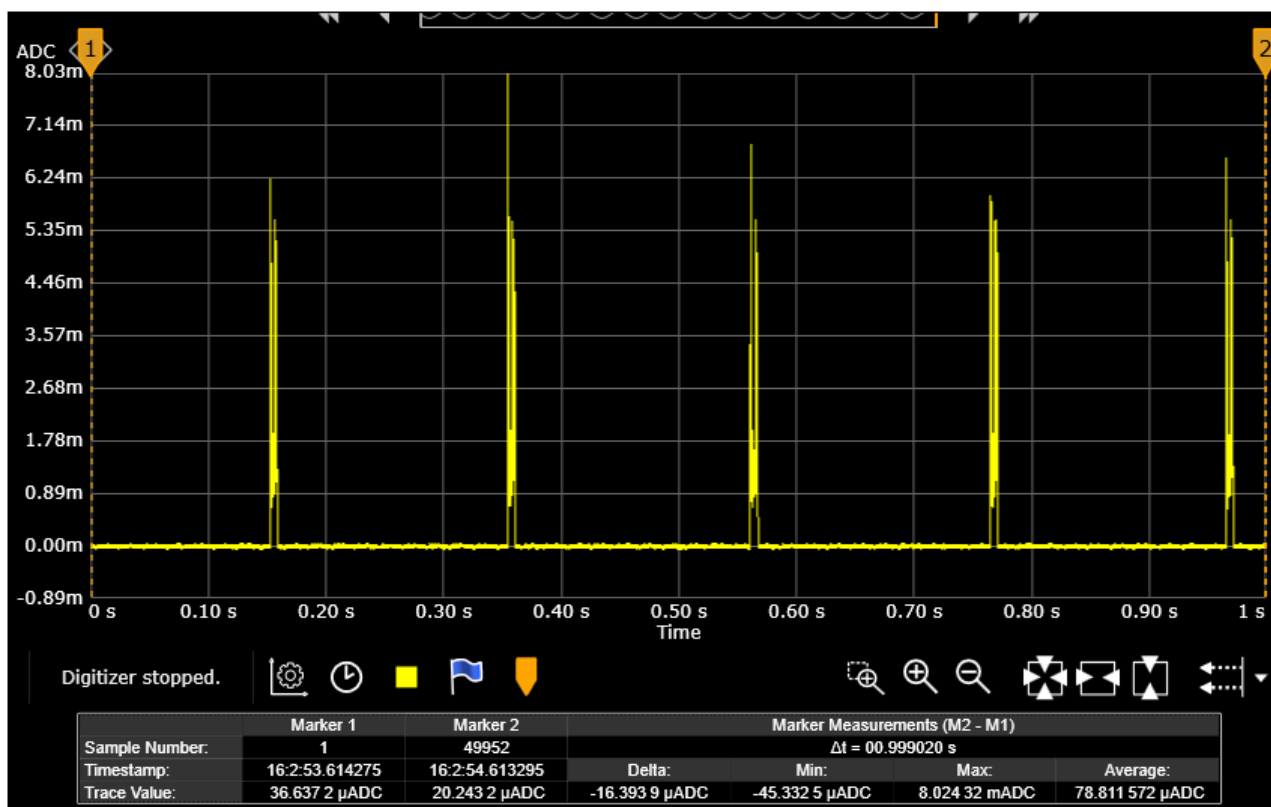
- 启动后默认的 ADV 周期为 200ms，由于 Inquiry Scan 和 Page Scan 也会自动打开，为测试 BLE 的功耗，需要使用 `btskey` 命令关闭 Scan，关闭命令参考经典蓝牙的 Scan 部分。例如，先发送 `btskey s` 命令，如果显示当前位于主菜单，就可以依次发送以下三个命令关闭 Page Scan 和 Inquiry Scan。发送完 `btskey 0`，可以再发送 `btskey 4` 查询 Scan 的状态。

- `btskey 1`
- `btskey 7`
- `btskey 0`

- 将唤醒 PIN 接高电平，系统进入低功耗模式，电流值如下图明显的下降，测量 200ms 间隔时的电流，低功耗模式下的电流波形见 ADV=200ms 的电流波形图，记 10 秒的平均电流为 C1，两个峰之间的电流记为睡眠电流 C2，ADV 的增量电流 $C=C1-C2$



进入低功耗模式电流变化



ADV=200ms 的电流波形

1. 将唤醒 PIN 接低电平，系统退出低功耗模式，在 console 里发送命令 `ble_config adv 500`，将 ADV 间隔改为 500ms
2. 将唤醒 PIN 接高电平，系统再次进入低功耗模式，测量 500ms 间隔时的电流

重复步骤 5 和 6，测量 ADV 间隔为 1000ms 的电流。

3.5. 连接场景

1. 打开串口调试工具，连接 HCPU 的 console 串口，连接测量设备与被测模块
2. 唤醒 PIN 接低电平，按底板的 Reset 键复位，启动成功后出现如下图的 log。类似 ADV 章节量 ADV 电流，也需要关闭 BT Scan

```

to /
- SiFLi Corporation
  / build Jan 21 2021, v0.6.2
  / 2020 - 2021 Copyright by SiFLi team
Patch hook install..
[1326] D/DS: data service subscribe (BLE_NV), service (0)
[32m[1355] I/nvds: Wait nvds service.
[0m[1379] D/DS: data service subscribe (ANCS), service (2010b6ac)
[1411] D/DS: data service subscribe dev open.
[32m[1464] I/ble_cm: read_bond_infor_from_flash: 7
[0m[32m[1502] I/sibless: enable BLE Core..
[0m[32m[1525] I/ble_app: status 0
[0m[32m[1548] I/nvds: nvds_msg_id 3841
[0m[32m[1571] I/nvds: get_msg 0
[0m[32m[1590] I/nvds: Subscribed nvds service ret 0.
[0m[32m[1624] I/ancs_srv: ancs event handler 116

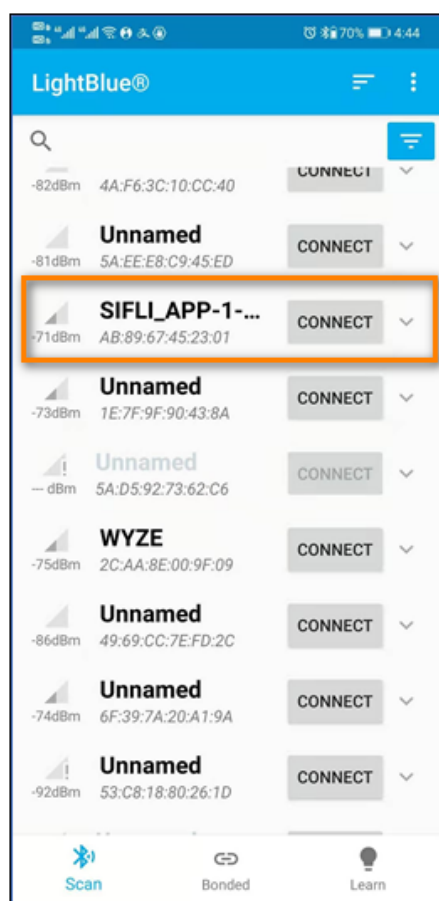
[0m[32m[1650] I/ble_cm: BLE_NVDS_ASYNC_FLUSH_TAG_CNF
[0m[32m[1678] I/ancs_srv: ancs event handler 112

[0m[32m[1704] I/ble_cm: BLE_NVDS_ASYNC_READ_CNF
[0m[31m[1728] E/ble_cm: len check error! 210, 0
[0m[32m[1753] I/ble_cm: BLE_NVDS_ASYNC_READ_CNF 0x10
[0m[32m[1780] I/ble_cm: BLE_NVDS_ASYNC_READ_CNF 0x1
[0m[32m[1806] I/ble_cm: BLE_NVDS_ASYNC_READ_CNF 0x17
[0m[32m[19753] I/sibless:
BLE ready!

[0m[32m[19777] I/ancs_srv: ancs event handler 0

```

1. 在手机打开如下图的 LightBlue 软件，在 Scan 列表找到名为 SIFLI_APP 的设备，点击 CONNECT 连接设备



1. 连接成功后 BLE 进入连接状态，默认的连接周期为 15ms
2. 将唤醒 PIN 接高电平，系统进入低功耗模式，测量 15ms 间隔时的电流
3. 将唤醒 PIN 接低电平，系统退出低功耗模式，在 console 里发送命令 `ble_config conn 50`，将连接周期改为 50ms，确认 console 出现如下图打印，其中 Updated connection interval: 40 表示 1.25ms 单位的连接间隔， $40 \times 1.25 = 50\text{ms}$ ，如果未出现打印，则表示参数更新失败，需要再次发送命令，同时观察电流波形确认间隔是否更新成功

```
msh >
msh >ble_config conn 50|
[0m[32m[2017749] I/sibiles: GAP event 3584
msh >[0m[32m[2017772] D/BLE_GAP: GAPC event status, op: 9, ret 0
[32m[2017798] I/ancs_srv: ancs event handler 63
[0m[32m[2017827] I/sibiles: GAP event 3601
[0m[32m[2017850] I/ble_app: Updated connection interval :40
[0m[32m[2017880] I/ancs_srv: ancs event handler 62
[0m[32m[2017908] I/ble_cm: BLE_GAP_UPDATE_CONN_PARAM_IND
[0m[32m[2017937] I/ble_cm: connection_manager_event_process 0x49
[0m[32m[2017970] I/ancs_srv: ancs event handler 170
```

1. 将唤醒 PIN 接高电平，系统再次进入低功耗模式，测量 50ms 周期时的电流，与 ADV 电流类似，测量 10秒的平均电流 C1，两个峰之间的电流记为睡眠电流 C2，连接状态的增量电流即为 $C=C1-C2$

重复步骤 6 和 7，可以测量连接周期为 200ms、500ms 和 1000ms 时的电流。

3.6. 测量结果汇总

表格中 ADV 与 Connection 的电流值为该模式的增量电流，Sleep 为睡眠电流，实际的平均电流等于增量电流与睡眠电流之和，例如 1 秒间隔的 ADV，4dBm 的发射功率，平均电流为 $10.5+14.5=24.5\mu A$

3.6.1. BLE 功耗表

模式	条件	电源电压 @TXpower=0dBm	3.3V	电源电压 @TXpower=4dBm	3.3V	电源电压 @TXpower=10dBm	3.3V	电源电压 @TXpower=13dBm	3.3V	单位
ΔBLE ADV	200ms	46		52.9		70.5		239.3		uA
	500ms	18.4		20.8		29.8		103.5		uA
	1s	8.9		10.5		14.6		53		uA
ΔBLE Connection	200ms	21		22		25.7		46.8		uA
	500ms	9		9.3		10.9		19.3		uA
	1s	5		5		6.1		10.3		uA
Sleep		14.5		14.5		14.5		14.5		uA

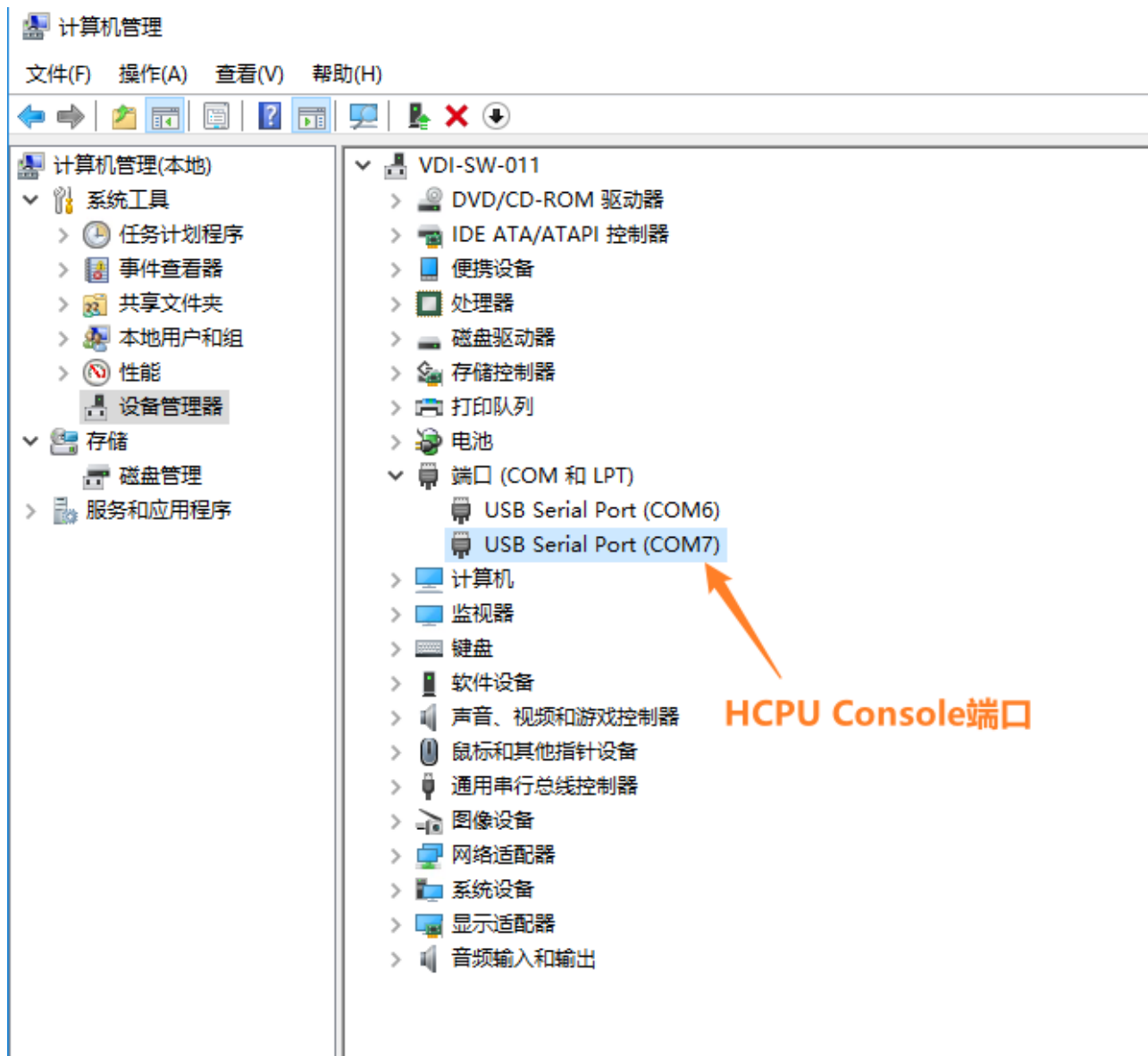
4. 经典蓝牙功耗测试

经典蓝牙 (Bluetooth Classic) 提供了比 BLE 更高的数据传输速率，适用于音频传输、数据同步等应用场景。本章详细介绍经典蓝牙在各种工作模式下的功耗特性。

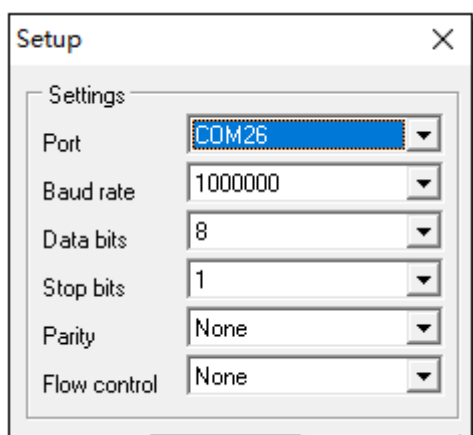
4.1. 测试说明

使用 BT 例程可以测试 Scan 和 Sniff 模式的功耗，系统上电后测试例程自动开启 Scan 和 ADV，使用手机连接蓝牙设备，在 HCPU 的 console 里可以发送命令修改配置，发送的命令都需以回车换行结尾。

PC与调试板使用 USB Type-C线连接后会枚举出两个串口，其中 HCPU使用第二个串口作为 console端口，如下图所示。



串口设置参见下图，波特率均设置为 1000000。



为便于控制测试条件，使用 PA24作为 HCPU的唤醒 PIN。当唤醒 PIN为低电平时 HCPU无法进入低功耗模式，此时可以通过 console给 HCPU发送命令修改参数，当 PA24悬空或者接高电平（即 3.3V电压，下文如未特别

说明，高电平均指 3.3V电压）时，HCPU进入低功耗模式，LCPU则周期性的进出低功耗模式，此时 console无法使用。

测试例程 LCPU的主频为 24MHz，HCPU的低功耗模式为 Deepsleep，LCPU的低功耗模式为 Standby。HCPU使用 **btskey**命令操作菜单修改配置。

4.2. 例程编译与烧录

4.2.1. 编译

进入example\pm\bt\project\hcpu目录，执行

```
scons --board=sf32lb52-core_n16r16 -j8
```


编译生成HCPU的image文件，编译生成的

image文件保存在

build目录下。

... example > pm > bt > project > hcpu > build_sf32lb52-core_n16r16_hcpu >				
↑↓ 排序 ≡ 查看 ...				
名称	修改日期	类型	大小	
board	2025/9/22 18:04	文件夹		
bootloader	2025/9/22 18:04	文件夹		
ftab	2025/9/22 18:05	文件夹		
lcpu	2025/9/22 18:05	文件夹		
main	2025/9/22 18:04	文件夹		
sifli_sdk	2025/7/21 9:44	文件夹		
.config	2025/9/22 18:04	Configuration 源文件	23 KB	
.config.old	2025/9/22 17:52	OLD 文件	23 KB	
compile_commands.json	2025/9/22 18:04	JSON 源文件	2,697 KB	
cppdefines.txt	2025/9/22 18:04	文本文档	1 KB	
custom_mem_map.h	2025/9/22 18:04	C Header 源文件	3 KB	
download.bat	2025/9/22 18:05	Windows 批处理文件	1 KB	
download.jlink	2025/9/22 18:05	JLINK 文件	1 KB	
ImgBurn.log	2025/9/22 18:07	文本文档	5 KB	
ImgBurnList.ini	2025/9/22 18:05	配置设置	1 KB	
Kconfig	2025/9/22 18:04	文件	1 KB	
kconfiglist	2025/9/22 18:04	文件	10 KB	
link.sct	2025/9/22 18:04	Windows Script Co...	6 KB	
main.asm	2025/9/22 18:05	ASM 文件	18,264 KB	
main.axf	2025/9/22 18:05	AXF 文件	8,373 KB	

4.2.2. 烧写镜像

在命令行编译的目录下执行

```
build_sf32lb52-core_n16r16_hcpu\uart_download.bat
```

烧写 build目录下编译生成的镜像文件。

4.2.3. 改变发射功率

工程默认配置的发射功率是0dbm，可以使用 `ble_tx_pwr_save x` 命令修改发射功率，x是需要修改的发射功率大小，例如改变发射功率为10dbm：`ble_tx_pwr_save 10`。改完后会自动重启生效。

4.3. 设备初始化

第一次使用开发板测试蓝牙需要在上电后执行以下初始化命令，命令在 HCPU的 console中执行，唤醒 PIN需接低电平避免 HPSYS进入低功耗模式，否则 HCPU无法处理命令

1. `nvds reset_all 1`

2. `nvds update addr 6 <蓝牙地址>`

例如 `nvds update addr 6 1234567890C8`，注意蓝牙地址分类需要保持一定的格式，建议 C8不要改动，其他的用户可以自行改变，命令执行完成后按 **Reset**键重启设备

4.4. Scan

1. 。打开串口调试工具，连接 HCPU的 console串口，连接测量设备与被测模块

2. ◦ 唤醒 PIN接低电平，复位开发板，启动成功后出现如下图的

log

```

\ \ /
- Si
[20:18:56.320]收←◆Fli Corporation
/ \ \ build on Oct 28 2022, 2.0.4 build cda680
2020 - 2022 Copyright by SiFli team
mount /dev success

[20:18:57.747]收←◆[1] I/drv.rtc main: PSCLR=0x80000100 DivAI=128 DivAF=0 B=
[1] I/drv.rtc main: Init RTC, wake = 0

[134] I/drv.audproc main: init 00 ADC_PATH_CFG0 0x924

[134] I/drv.audproc main: HAL_AUDPROC_Init res 0

[136] I/drv.audcodec main: HAL_AUDIOCODEC_Init res 0

[140] I/audio main: Device audcodec opened[handle = 20031e64]

call par CFG1(35bb)
fc 9, xtal 2000,
[20:18:57.803]收←◆pll 2044
call par CFG1(35bb)
fc 9, xtal 2000, pll 2044
mbox_stack:20020000,1024
proc_stack:20020400,4096
Register mpi3 to mtd device with base addr 0x64300000
disk_init:0
init
mount fs on mpi3 to root success
ble db init: sector_size 4096 size 16384
[689] I/nvds main: read sleep time 4500
[694] D/DS main: data service subscribe (BT_EN), service (0)
[698] I/mw.sys ds_proc: turn on lcpu
msh />[40645] I/NO_TAG ds_proc: BT enable LCPU 0.

[20:18:57.831]收←◆[42814] I/NO_TAG ds_proc: BT enabled result 0

[20:18:58.562]收←◆
#####
##                               ##
##      BTS2 Demo Main Menu      ##
##      1. Generic Command        ##
##      2. SPP Client             ##
##      3. SPP Server             ##
##      4. HFP HF                 ##
##      6. A2DP Sink              ##
##      p. AVRCP                  ##
##      s. Show Menu              ##
##      q. Exit                   ##
##                               ##
#####
ble db init: sector_size 4096 size 16384
[66439] I/nvds main: read sleep time 4500
[66925] I/ble_cm main: read_bond_infor_from_flash: 0
[66926] I/sibles main: enable BLE Core..

[20:18:59.155]收←◆<< Local device address: CDAB:78:563412
>> SPP enabled
>> Handfree enable success
>> AVCTP enabled
>> Headset enable success
<< Local LMP version: 12
<< Local LMP subversion: 0
<< Local controller manufacturer name id: 2636
<< Local device name changed!
<< Class of device has been changed!
<< Write scan enable success
<< Local device name: SifliDemo-12-34-56-78-ab-cd

```

3. ◦ 启动后 ADV、Inquiry Scan和 Page Scan都会打开，为测试 Scan的功耗，需先关闭 ADV，关闭命令为

diss adv_stop

1. ◦ 开机后默认处于 BTS主菜单，通过发送 **btskey**命令可以配置打开或关闭 Scan。发送**btskey s**命令显示当前菜单，再按菜单提示发送命令进入子菜单。比如，在主菜单下，依次发送如下三个命令可以打开 Page Scan并关闭 Inquiry Scan

- (a) btskey 1
- (b) btskey 7
- (c) btskey 2

```
[21:29:38.944]发->◇btskey s
□
[21:29:38.971]收<-◆
#####
##
##          BTS2 Demo Main Menu          ##
##  1. Generic Command                    ##
##  2. SFP Client                        ##
##  3. SFP Server                        ##
##  4. HFP HF                            ##
##  6. A2DP Sink                          ##
##  p. AVRCP                            ##
##  s. Show Menu                          ##
##  q. Exit                              ##
##                                         ##
#####

btskey s
s

msh />
msh />
[21:29:47.349]发->◇btskey 1
□
[21:29:47.368]收<-◆
#####
##
##          Generic Command Menu          ##
##  1. Inquiry start                      ##
##  2. Inquiry cancel                    ##
##  3. Select device from Inquiry list    ##
##  4. Sc menu                           ##
##  5. Local device info                  ##
##  6. Get remote device info             ##
##  7. Scan mode                          ##
##  8. Link menu                          ##
##  9. Service Browse                     ##
##  a. Set IO Settings                    ##
##  s. Show Menu                          ##
##  r. Return to last menu                ##
##                                         ##
#####

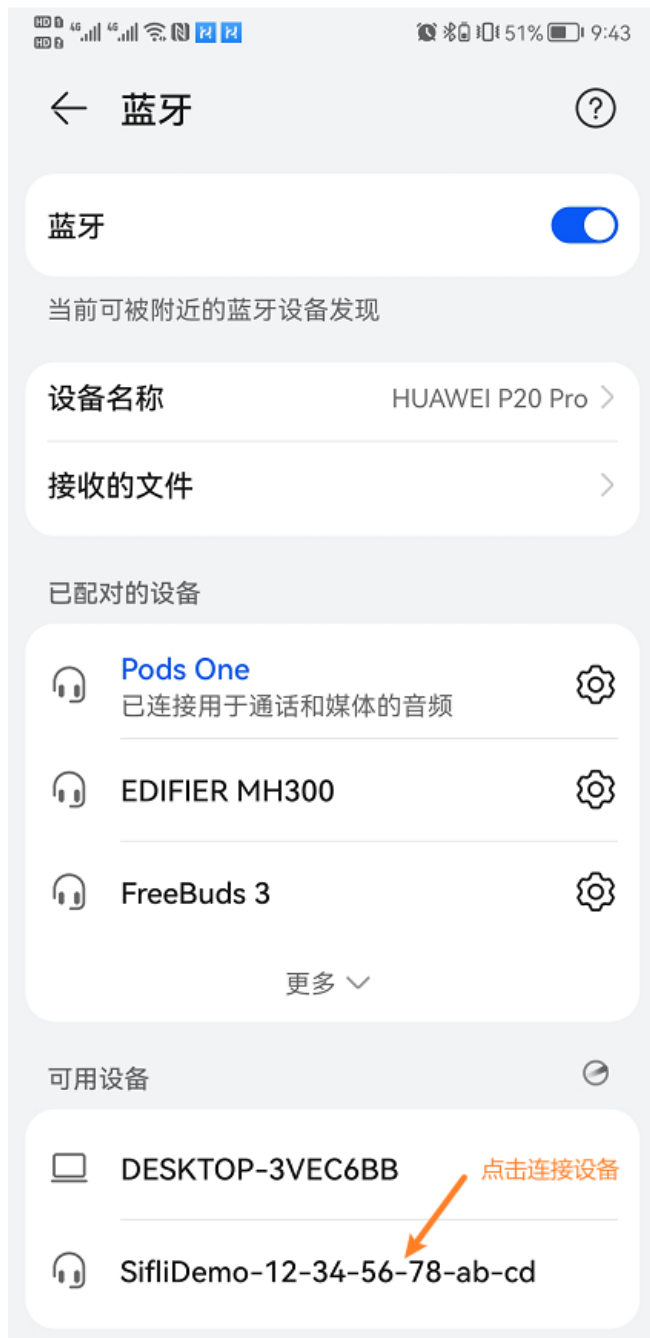
btskey 1
1

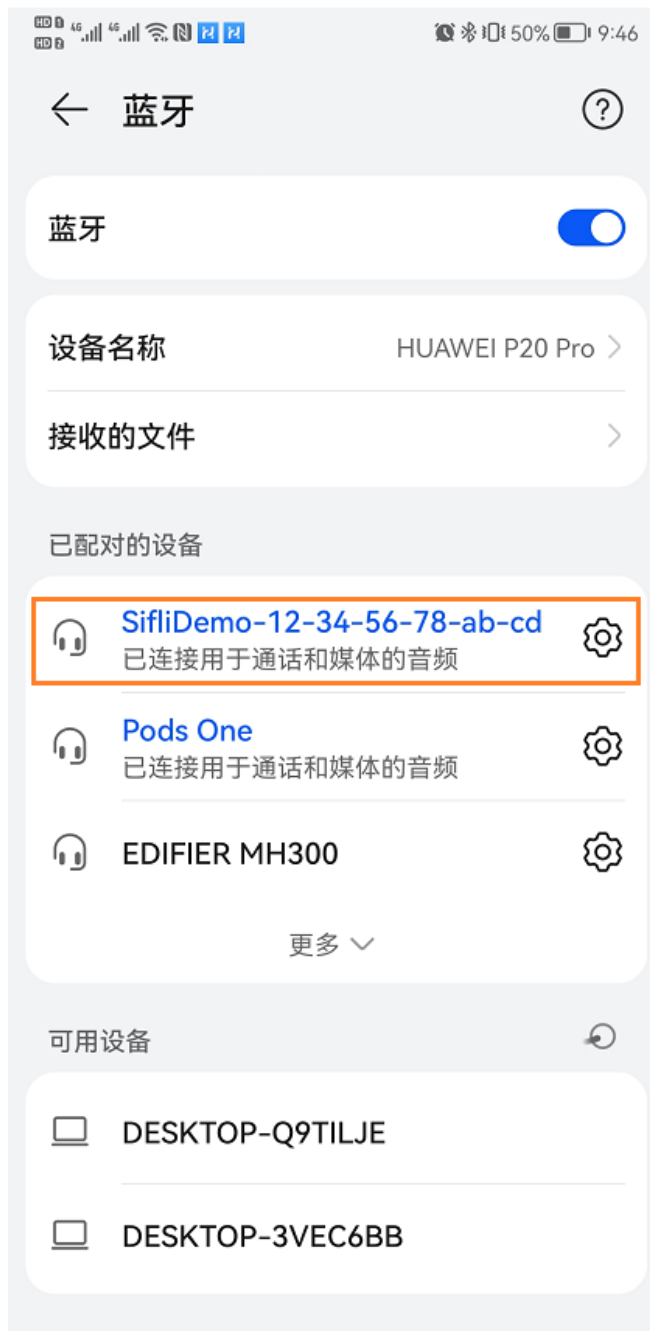
msh />
msh />
[21:30:05.706]发->◇btskey 7
□
[21:30:05.725]收<-◆
#####
##
##          Scan Mode Select Menu          ##
##  0. No scans enabled                    ##
##  1. Inquiry scan enabled and Page scan disabled ##
##  2. Page scan enabled and Inquiry scan disabled ##
##  3. Inquiry scan and Page scan enabled    ##
##  4. Read scan mode                      ##
##  s. Show Menu                          ##
##  r. Return to last menu                ##
##                                         ##
#####
```

1. 发送 BTS 命令配置设备只发送 Inquiry Scan 或者 Page Scan，唤醒 PIN 接高电平，系统进入低功耗模式，测量 1 分钟的平均电流，记为 Scan 的平均电流 C1，同时再测量两个峰之间的底电流，记为睡眠电流 C2，Scan 的增量电流即为 C=C1-C2。测试程序的 Page Scan 周期为 1.28 秒，Inquiry Scan 周期为 2.56 秒，所以 Inquiry Scan 的增量电流为 Page Scan 的一半。
2. 唤醒 PIN 接低电平，发送 BTS 命令配置设备同时发送 Inquiry Scan 和 Page Scan，再把唤醒 PIN 接高电平，系统进入低功耗模式，测量测试的平均电流，记为 Both Scan 的电流

4.5. Sniff Mode

1. ◦ 参考Scan章节的步骤复位开发板并关闭 ADV
2. ◦ 手机连接开发板对应的蓝牙设备，点击后弹出配对窗口，点确认后配对成功，蓝牙设备进入已配对设备列表，并显示已连接状态,如下图：





1. ◦ 连接完成后，等待一会儿，console 窗口会打印 »Sniff mode，表示设备已进入 Sniff 模式
2. ◦ 发送 `btskey` 命令关闭 Scan，具体步骤如下：

- (a) `btskey s`：查看当前菜单
- (b) 如果是 HFP HP Menu，则发送
- (c) `btskey r`：返回上一级菜单 BTS2 Demo Main Menu，再顺序发送以下命令
- (d) `btskey 1`：选择 Generic Command

- (e) btskey 7: 选择 Scan mode
(f) btskey 0: 关闭 scan

1. 与 Scan 电流的测量方法类似，Sniff 模式的增量电流可以由 10 秒的平均电流与两个峰之间的睡眠电流相减得到

4.6. 测量结果汇总

表格中 Sniff 模式和 Scan 的电流值为该模式的增量电流，Sleep 为睡眠电流，实际的平均电流等于增量电流与睡眠电流之和，例如 Sniff 模式下，500ms 间隔，attempt=4，10dBm 的发射功率，平均电流为 $23+16.6=39.6\mu A$

4.6.1. BT 功耗

模式	条件	电源电压 @TXpower=0dBm	3.3V	电源电压 @TXpower=4dBm	3.3V	电源电压 @TXpower=10dBm	3.3V	电源电压 @TXpower=13dBm	3.3V	单位
ΔBT Sniff Mode	500ms (attempt=4)	18.6		23.4		23		40.9		uA
ΔBoth Scan	Inquiry Scan and Page Scan	52.6		52.6		52.6		52.6		uA
Sleep	16.6	16.6		16.6		16.6		16.6		uA

1. Inquiry Scan 每 2.56s 接收 16.87ms，Page Scan 每 1.28s 接收 16.87ms。

2. bt 500ms sniff@TXpower10dBm: $=23+16.6=39.6\mu A$

5. 处理器性能与功耗测试

本章节专注于 SF32LB52x 处理器在不同计算负载下的功耗特性分析，包括标准化基准测试和基础循环测试。

5.1. 测试目标

通过标准化的处理器测试，评估芯片在不同计算强度下的功耗表现：

- Coremark 基准测试：业界标准的 CPU 性能基准测试
- While Loop 测试：基础循环操作的功耗基线测试
- 关机模式测试：最低功耗状态的功耗测量

5.2. 测试说明

CoreMark 例程测试以下场景下的功耗：

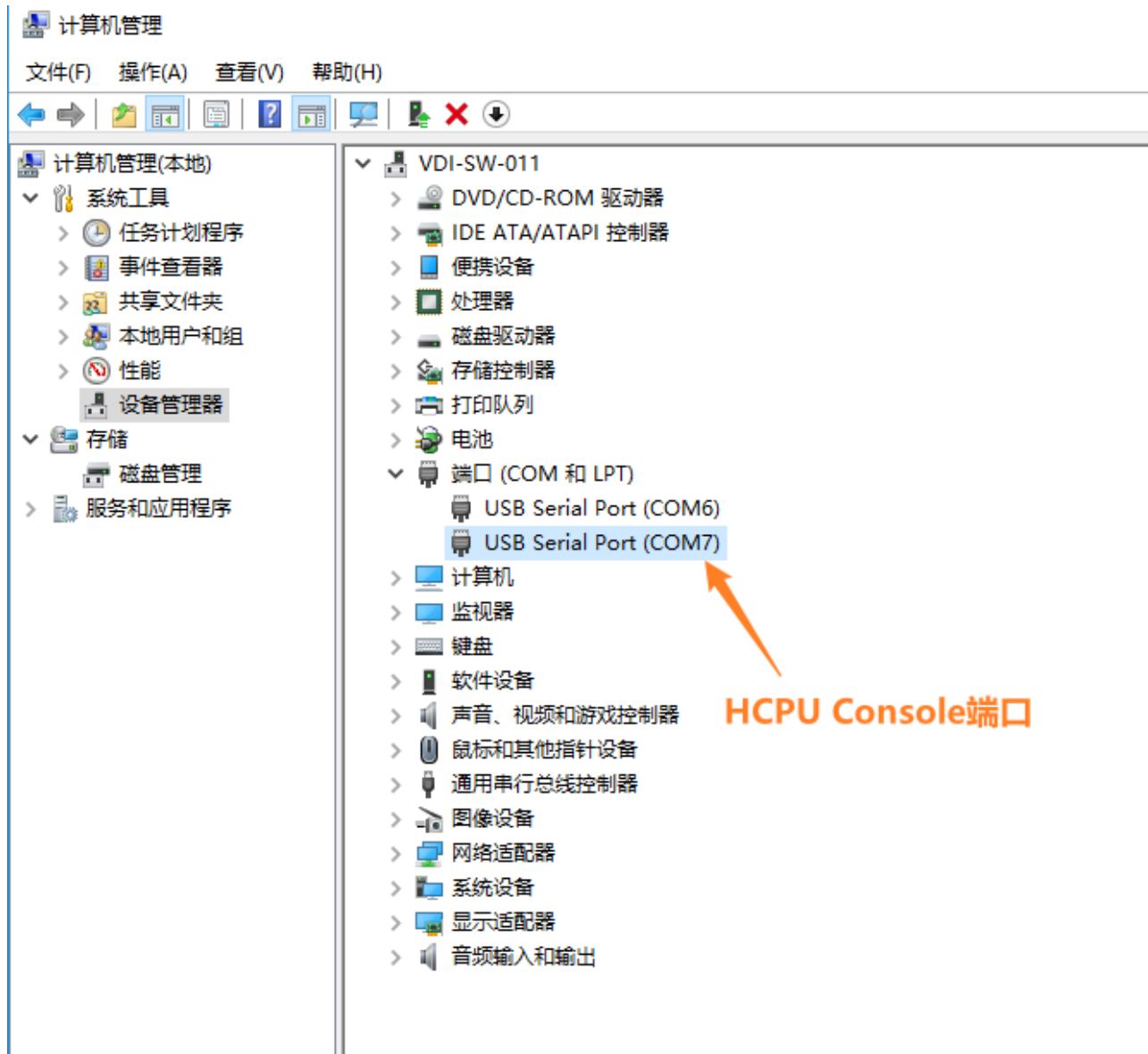
- 一个核执行 CoreMark 基准测试程序
- 一个核执行一段时间的 while 循环，循环中执行 nop 指令
- 系统关机，可由 RTC 定时唤醒
- 系统关机，由按键唤醒

LCPU 执行测试命令时 HCPU 自动进入 DeepSleep 低功耗模式，这期间 HCPU 无法响应 console 命令，测试完成后 HCPU 退出低功耗模式，此时可以继续发送新的测试命令，如果 LCPU 测试一直没有结束，可以将 PA30 接低电平，强制唤醒 HCPU，但需要注意，测量电流时，PA24 不能接低电平，必须接高电平或者悬空，否则会导致测量值偏高。

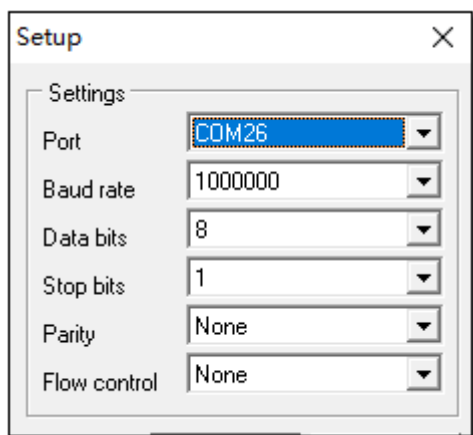
LCPU 始终不进入低功耗模式，当不执行任务时 LCPU 处于 WFI 状态。

需要注意发送的命令都要以回车换行符结尾。

PC 与调试板使用 USB Type-C 线连接后会枚举出两个串口，HCPU 使用第二个串口作为 console 端口，如下图所示。



串口设置参见下图，波特率均设置为 1000000。



HCPU 可使用以下命令启动相应任务：

- `run_coremark freq_in_mhz`: 修改主频并执行 CoreMark, `freq_in_mhz`是以 MHz 为单位的频率 例如：
`run_coremark 48`, 以 48MHz 主频执行 CoreMark
- `run_while_loop freq_in_mhz`: 修改主频并执行一段时间 while loop, `freq_in_mhz`是以 MHz 为单位的频率
例如: `run_while_loop 48`, 以 48MHz 主频执行 while loop

使用 HCPU 的 console 发送以下命令，可以让 LCPU 执行指定任务：

- `tolcpu run_coremark freq_in_mhz`
- `tolcpu run_while_loop freq_in_mhz`

命令含义同 HCPU。HCPU 还支持以下命令：

- `shutdown [wakeup_time_in_sec]`: 关机，`wakeup_time_in_sec`为可选参数，单位为秒，表示关机后多久自动开机，如果不带参数，则关机后只能被 KEY1 按键唤醒

注：由于编译采用 Ofast 优化等级，跑分结果仅作参考，并不能达到 Omax 时的最高分。

5.3. 例程编译与烧录

5.3.1. 编译

如果使用已编译好的 `image` 文件，可以直接跳到烧录部分进行烧录开始测试。 进入 `example\pm\coremark\project\hcpu`目录，执行

```
scons --board=sf32lb52-core_n16r16 -j8
```

编译生成HCPU的image文件，编译生成的 image文件保存在 build 目录下。

+ ... pm > coremark > project > hcpu > build_sf32lb52-core_n16r16_hcpu > 在 build_sf32lb52-core_n16r16_hcpu 中				
排序 查看				
名称	修改日期	类型	大小	
board	2025/9/23 10:40	文件夹		
bootloader	2025/9/23 11:23	文件夹		
ftab	2025/9/23 11:23	文件夹		
lcpu	2025/9/23 11:23	文件夹		
main	2025/9/23 11:23	文件夹		
sifli_sdk	2025/7/23 18:12	文件夹		
.config	2025/9/22 20:30	Configuration 源文件	20 KB	
.config.old	2025/9/22 17:30	OLD 文件	20 KB	
compile_commands.json	2025/9/23 11:23	JSON 源文件	1,713 KB	
cppdefines.txt	2025/9/23 11:23	文本文档	1 KB	
custom_mem_map.h	2025/9/23 11:23	C Header 源文件	1 KB	
download.bat	2025/9/23 11:23	Windows 批处理文件	1 KB	
download.jlink	2025/9/23 11:23	JLINK 文件	1 KB	
ImgBurn.log	2025/9/23 11:23	文本文档	5 KB	
ImgBurnList.ini	2025/9/23 11:23	配置设置	1 KB	
Kconfig	2025/9/23 11:23	文件	1 KB	
kconfiglist	2025/9/23 11:23	文件	10 KB	
linker	2025/9/23 11:23	Windows Script Co	1 KB	

5.3.2. 烧录镜像

在命令行编译的目录下执行

```
build_sf32lb52-core_n16r16_hcpu\uart_download.bat
```

烧写 build 目录下编译生成的镜像文件。

5.4. CoreMark

5.4.1. HPSYS

1. 。打开串口调试工具，连接 HCPU 的 console 串口，连接测量设备与被测模块

2. ◦ 复位，启动成功后在 HCPU 的 console 中出现如下图的 log

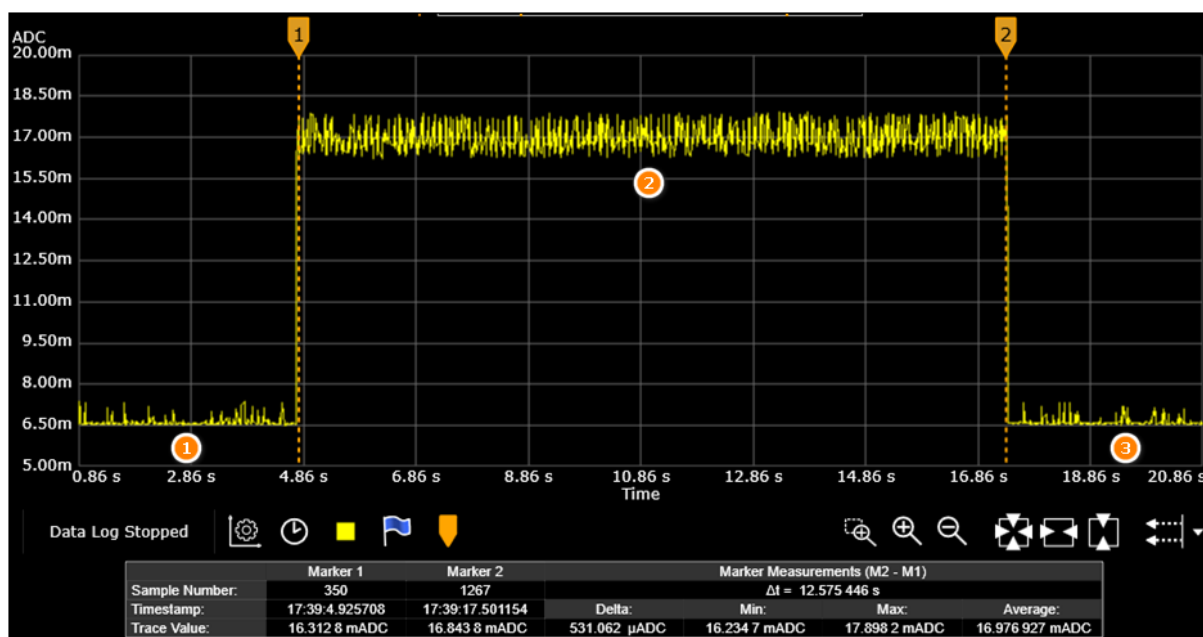
```

\OSerial: c2, Chip: 1, Package: 0, Rev: 0
- SiFli Corporation
  / build Jun 7 2021, v0.9.1.1-290-g6c1494aad
2020 - 2021 Copyright by SiFli team
msh > [134] I/TOUCH main: Regist touch screen driver, probe=10038041
  
```

1. ◦ 发送命令run_coremark 192得到平均电流C1，发送run_coremark 168得到平均电流C2，得到192MHz和168MHz的增量电流 $C=(C1-C2)/(192-168)$
2. ◦ 发送命令run_coremark 144得到平均电流C1，发送run_coremark 120得到平均电流C2，得到144MHz和120MHz的增量电流 $C=(C1-C2)/(144-120)$
3. ◦ 发送命令run_coremark 48得到平均电流C1，发送run_coremark 24得到平均电流C2，得到48MHz的增量电流 $C=(C1-C2)/(48-24)$
4. ◦ 发送命令run_coremark 24得到平均电流C1，发送run_coremark 12得到平均电流C2，得到24MHz和12MHz的增量电流 $C=(C1-C2)/(24-12)$

• 注意：24MHz有两种档位：D0和D1。默认是使用D0档位跑24Mhz。如果需要测量D1档位，可以在命令后面带个参数1表示用D1档跑24M： `run_coremark 24 1`

1. ◦ 如下图所示，阶段 1 是 HCPU 跑在 192MHz 主频时 WFI 模式下的电流波形，开始执行 CoreMark后进入阶段 2，电流上升并保持至测试结束，阶段 3 为回到 WFI 模式的电流波形



48MHz/24MHz/12MHz 等频率下 coremark 执行时间会很长，不必等待测试完成，测量得到电流值后可以复位测试下一项。

5.5. While Loop

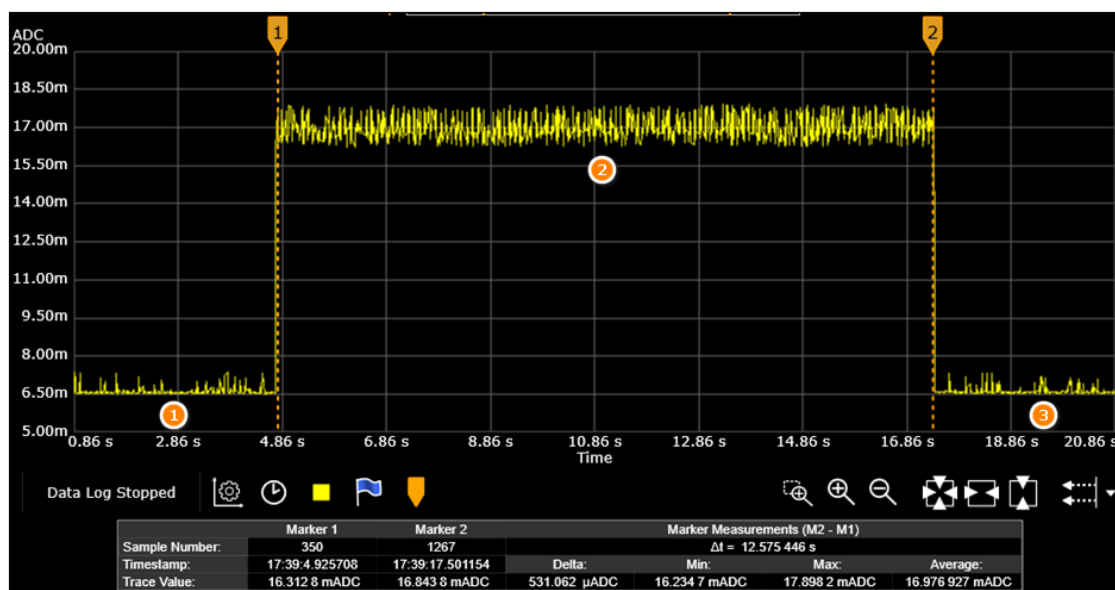
5.5.1. HPSYS

1. 打开串口调试工具，连接 HCPU 的 console 串口，连接测量设备与被测模块
2. 复位，启动成功后在 HCPU 的 console 中出现如下图的 log

```

\OSerial:c2, Chip:1, Package:0, Rev:0
\\
- SiFli Corporation
// build Jun 7 2021, v0.9.1.1-290-g6c1494aad
2020 - 2021 Copyright by SiFli team
msh >[134] I/TOUCH main: Regist touch screen driver, probe=10038041
    
```

1. 发送命令run_while_loop 192，指示 HCPU 在 192MHz 频率下执行 while loop 测试，测量此时的平均电流C1，如下图HCPU While Loop 测试电流图所示，阶段 1 是 HCPU 跑在 192MHz 主频时 WFI 模式下的电流波形，开始执行 while loop 后进入阶段 2，电流上升并保持至测试结束，阶段 3 为回到 WFI 模式的电流波形。console 中显示 HCPU 的频率和 while loop 的持续时间，如图HCPU While Loop Log图所示



HCPU While Loop 测试电流图

```
\0Serial:c2,Chip:1,Package:b,Rev:0
- SiFli Corporation
  build Jun 16 2021, v0.9.3-3-gf1419402d
  2020 - 2021 Copyright by SiFli team
msh >[783] I/TOUCH main: Regist touch screen driver, probe=1003948d
msh >run_while_loop 240
Current HCPU freq: 240000000
New HCPU freq: 240000000
Start
Done. Run for 5.38 seconds
msh >
```

HCPU While Loop Log 图

1. 发送命令run_while_loop 168，指示 HCPU 在 168MHz 下执行 while loop 测试，测量此时的平均电流 C2，由此得到每 MHz 的电流增量为 $C=(C1-C2)/(192-168)$
2. 执行run_while_loop 144和run_while_loop 120，测量 144MHz 和 120MHz 的平均电流，得到 144MHz 的增量电流 $C=(C1-C2)/(144-120)$
3. 执行run_while_loop 48和run_while_loop 24，测量 48MHz 和 24MHz 的平均电流，得到 48MHz 的增量电流 $C=(C1-C2)/(48-24)$
4. 执行run_while_loop 24和run_while_loop 12，测量 24MHz 和 12MHz 的平均电流，得到 24MHz 的增量电流 $C=(C1-C2)/(24-12)$

5.6. 关机

1. 打开串口调试工具，连接 HCPU 的 console 串口
2. 复位，启动成功后在 HCPU 的 console 中出现如下图的 log

```
msh >
msh >shutdown
Shutdown, press key1 to wake up system...
msh >
msh >\0Serial:c2,Chip:1,Package:b,Rev:0msh >[0] I/TOUCH main: Regist touch screen driver, probe=1003948d
msh >
msh >
msh >
```

关机

按KEY1后的开机log

3. 发送命令shutdown，指示系统关机，关机后只能被 KEY1 按键唤醒，由于关机电流比较小，建议使用万用表测量
4. 按 KEY1 键唤醒系统，console 中再次出现开机阶段的 log，但不会出现版本号的 log
5. 发送命令shutdown 5，如下图指示系统关机并在 5 秒后会自动开机，为了便于测量，可以将等待时间延长

```
msh >shutdown 5
Shutdown, wake up after 5 second...
msh >\0Serial:c2,Chip:1,Package:b,Rev:0msh >[0] I/TOUCH main: Regist touch screen driver, probe=1003948d
```


5.7. 测量结果汇总

5.7.1. 处理器功耗

5.7.1.1. CoreMark 测试结果

时钟频率	电源电压 3.3V 电流(uA)	增量 (uA/MHz)	CoreMark
240MHz	9380	35	240MHz
216MHz	8560		216MHz
192MHz	7760		192MHz
168MHz	6920		168MHz
144MHz	5350	27	144MHz
120MHz	4700		120MHz
48MHz	1720	30	48MHz
24MHz D0	936	32	24MHz D0
12MHz	549		12MHz
24MHz D1	1003	.	24MHz D1

5.7.1.2. WhileLoop 测试结果

时钟频率	电源电压 3.3V 电流(uA)	增量 (uA/MHz)
240MHz	6910	24
216MHz	6340	
192MHz	5790	
168MHz	5200	
144MHz	4080	25
120MHz	3470	
48MHz	1400	17
24MHz D0	806	27
12MHz	482	
24MHz D1	993	.

5.7.1.3. 关机功耗

测试项	电源电压 3.3V 电流(uA)
按键唤醒(发送 shutdown)	1.3

