SF32LB52x

Low Power Consumption Scenario Testing Report

latest (2026-06-04)

RPT5202

www.sifli.com

Table of Contents

1. SF32LB52x Chip Introduction 4
1.1. Chip Architecture Overview 5
1.2. Test Content Overview 5
1.3. Important Notice 6
1.3.1. Key Factors Affecting Power Consumption 7

2. Test Environment Configuration Guide 7
2.1. Related Hardware 8
2.2. Related Software 8
2.3. Power Measurement Method 8
2.3.1. How to Switch to an External Antenna 9

3. BLE Bluetooth Low Energy Test 12
3.1. Test Description 13
3.2. Routine Compilation and Programming 14
3.2.1. Compilation 14
3.2.2. Programming Image 15
3.2.3. Transmit Power Configuration 16
3.3. Device Initialization 16
3.4. ADV Scenario 16
3.5. Connection Scenario 18
3.6. Measurement Results Summary 20
3.6.1. BLE Power Consumption Table 21

4. Classic Bluetooth Power Consumption Test 21
4.1. Test Description 22
4.2. Routine Compilation and Programming 23
4.2.1. Compilation 23
4.2.2. Programming Image 24
4.2.3. Transmit Power Configuration 25
4.3. Device Initialization 25
4.4. Scan 25
4.5. Sniff Mode 29

4.6. Measurement Results Summary 32
4.6.1. BT Power Consumption 32

5. Processor Performance and Power Consumption Test 33
5.1. Test Objectives 34
5.2. Test Description 34
5.3. Routine Compilation and Programming 36
5.3.1. Compilation 36
5.3.2. Programming Image 37
5.4. CoreMark 37
5.4.1. HPSYS 38
5.5. While Loop 39
5.5.1. HPSYS 39
5.6. Shutdown 40
5.7. Measurement Results Summary 41
5.7.1. Processor Power Consumption 41

SF32LB52x Low Power Test Report

Welcome to the SF32LB52x series chip low power test report!

Chip Introduction

Introduction to the chip

Test Environment

Learn about the hardware environment, software configuration and testing tools required for testing

BLE Low Power Test

Detailed test methods and results for Bluetooth Low Energy scenarios

BT Classic Bluetooth Test

Power consumption test data for classic Bluetooth application scenarios

Coremark Performance Test

Power consumption performance analysis in benchmark performance testing

1. SF32LB52x Chip Introduction

1.1. Chip Architecture Overview

SF32LB52x is an innovative dual-core architecture SoC chip based on Arm Cortex-M33 STAR-MC1 processor.

High Performance Core

Specifically designed for high-performance computing scenarios:

- Graphics processing
- Audio processing
- Neural network computation

Low Power Core (LCPU)

Optimized for low power scenarios:

- Bluetooth protocol stack Controller layer
- Real-time task processing
- Standby state management

1.2. Test Content Overview

This document provides power consumption test methods and reference results for SF32LB52x series chips in various usage scenarios:

Test Scenario Overview

Category	Test Scenario	Description
Wireless Communication	BLE Bluetooth Low Energy	Typical scenarios such as advertising, connection, data transmission
Wireless Communication	BT Classic Bluetooth	Various application modes of traditional Bluetooth protocol
Performance Test	Coremark Benchmark Test	Power consumption performance in standardized CPU performance testing
Basic Test	While Loop	Power consumption baseline for basic loop operations
Standby Mode	Shutdown Mode	Power consumption data in the lowest power state

1.3. Important Notice

Warning

Test results are for reference only

The test results in this document are closely related to software and hardware environments, and differences may exist in actual applications.

1.3.1. Key Factors Affecting Power Consumption

Chip Factors

- Individual differences of the chip itself
- Manufacturing process batch variations

Hardware Factors

- Selection and quality of peripheral components
- External inductor for chip built-in Buck
- PCB design and routing

Test Environment

- Accuracy and calibration of test equipment
- Power supply stability and ripple
- Environmental temperature and humidity conditions

Software Configuration

- Firmware version and compilation options
- I/O pin configuration status
- Clock and power management settings

Environmental Conditions

- Operating temperature range
- Electromagnetic interference level
- Power supply voltage fluctuation

2. Test Environment Configuration Guide

Accurate power consumption testing requires standardized environment configuration. This chapter introduces various environmental requirements needed for testing.

2.1. Related Hardware

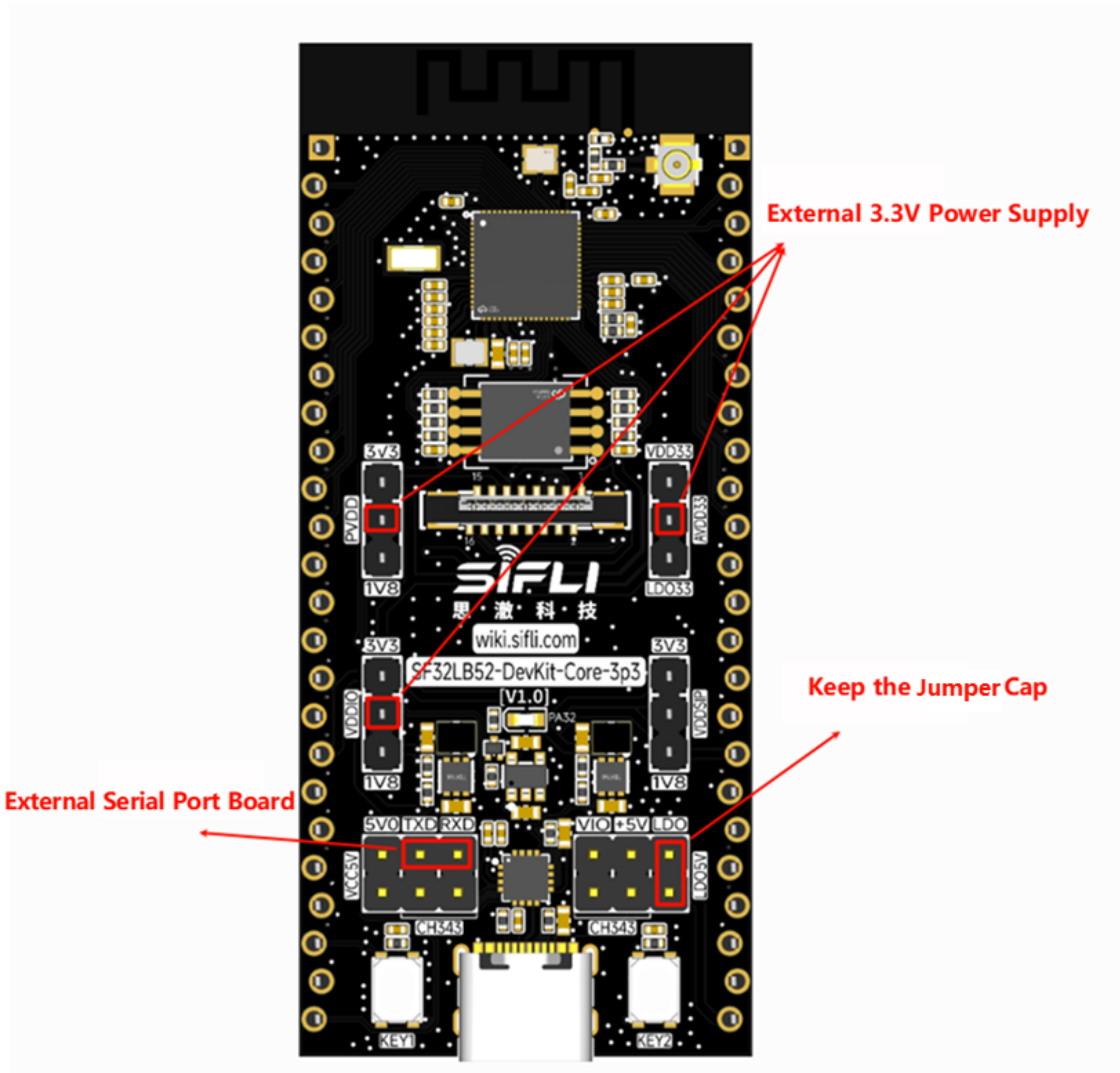
- SF32LB52-DevKit-Core-3p3 (on-board chip models SF32LB52BU56 / SF32LB52JUD6). For usage, refer to the SF32LB52-DevKit-Core-3p3 Development Board User Guide.
- Agilent 66319D DC power supply.
- Keysight 34465A digital ammeter.
- Serial (UART) board.
- Android phone, install the LightBlue App (com.punchthrough.lightblueexplorer_LightBlue_1.9.3.apk, available from the Google Play Store) or install a BLE Debug Assistant (available from the built-in Android app store).

2.2. Related Software

- BLE and BT power consumption test routines: example\pm\bt\src
- CoreMark power consumption test routines: example\pm\coremark\src软件

2.3. Power Measurement Method

Use a power measurement instrument to supply 3.3 V simultaneously to PVDD, AVDD33, and VDDIO. The supply pins are highlighted in the figure. Remove all other jumpers and keep the LDO5V jumper in place. Connect TXD and RXD to an external USB-to-UART board to input commands.

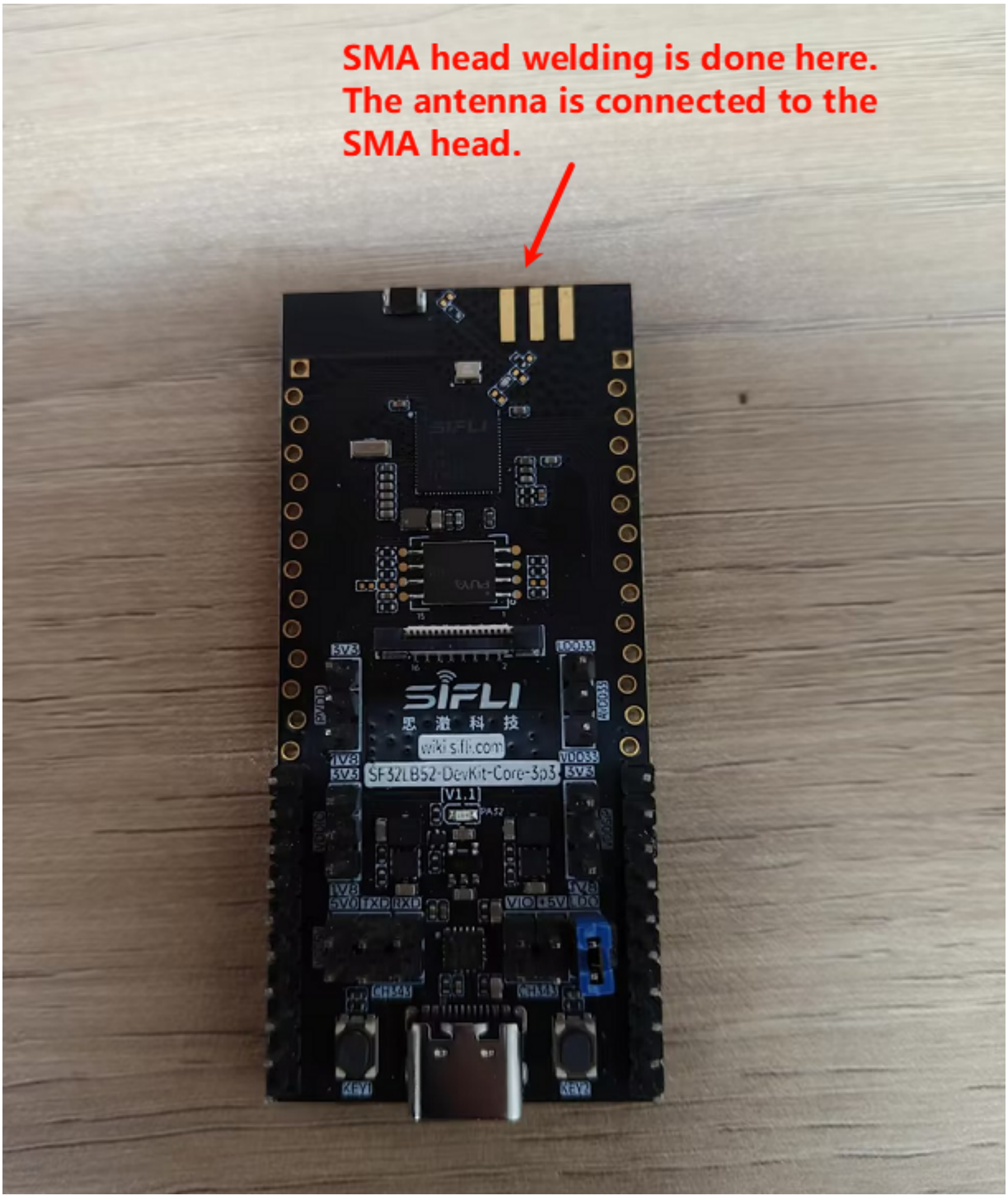


The development board uses the on-board antenna by default for RF transmit and receive. You may choose to switch to an external antenna.

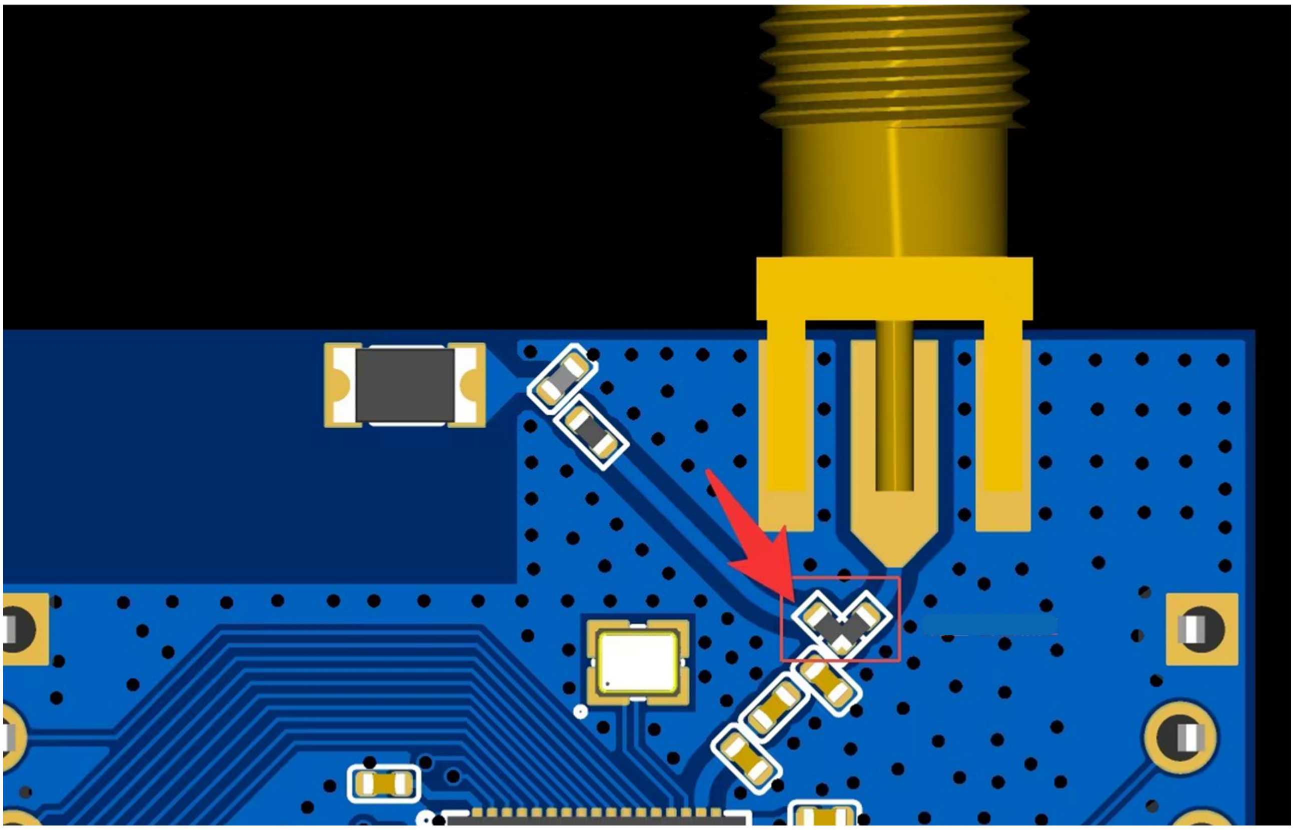
2.3.1. How to Switch to an External Antenna

1. Required hardware: SMA connector + antenna (or a 50-ohm load). Specification: 50-ohm RF load or 50-ohm RF antenna.

1. First, solder the antenna and SMA connector as shown below.



1. Then change the orientation of the selection resistor as shown to point to the external antenna. By default, it points to the on-board antenna.



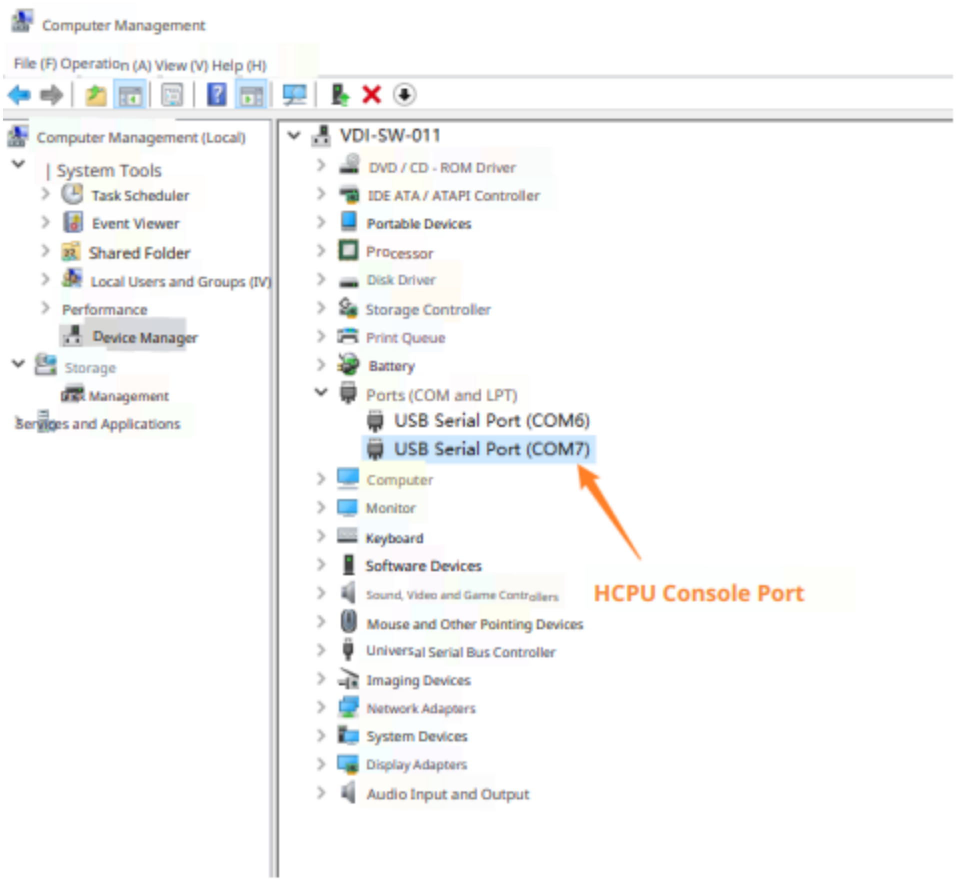
3. BLE Bluetooth Low Energy Test

BLE (Bluetooth Low Energy) is one of the core functions of SF32LB52x. This chapter provides detailed introduction to power consumption test methods and results in various BLE application scenarios.

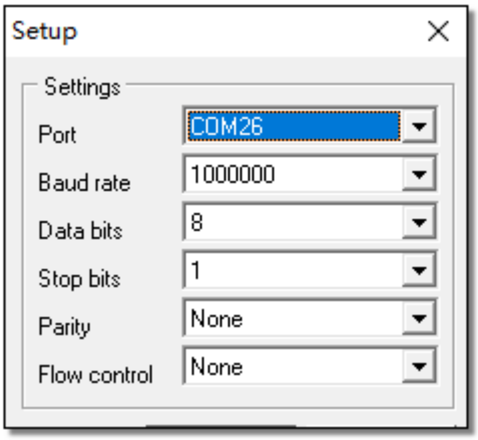
3.1. Test Description

Use BLE routine to test typical ADV and connection scenario power consumption. After system power-on, the routine automatically starts ADV transmission. Use the LightBlue software on your phone to discover and connect to the device. Send commands through the HCPU console to modify ADV transmission interval and connection period. All commands must end with carriage return and line feed characters.

After connecting the PC and debug board with USB Type-C cable, two serial ports will be enumerated, where HCPU uses the second serial port as the console port, as shown in the figure below.



Serial port settings are shown in the figure below: baud rate is set to 1000000.



For convenient control of test conditions, use PA24 pin as the HCPU wake-up PIN. When the wake-up PIN is at low level, HCPU cannot enter low power mode. At this time, commands can be sent to HCPU through console to modify parameters. When PA24 is floating or connected to high level (i.e., 3.3V voltage, high level refers to 3.3V voltage unless otherwise specified below), HCPU enters low power mode, and LCPU periodically enters and exits low power mode. At this time, console cannot be used.

In the test routine, LCPU main frequency is 24MHz, HCPU low power mode is Deepsleep, and LCPU low power mode is Standby.

HCPU supports the following commands to modify BLE ADV and connection period, and can also use btskey command to operate menu to modify BT configuration.

- ble_config adv interval_in_ms: where interval_in_ms is the ADV interval in milliseconds; For example: send command ble_config adv 100 to change ADV period to 100ms
- ble_config conn interval_in_ms: where interval_in_ms is the connection period in milliseconds; For example: after connection, send ble_config conn 100 to change connection period to 100ms

3.2. Routine Compilation and Programming

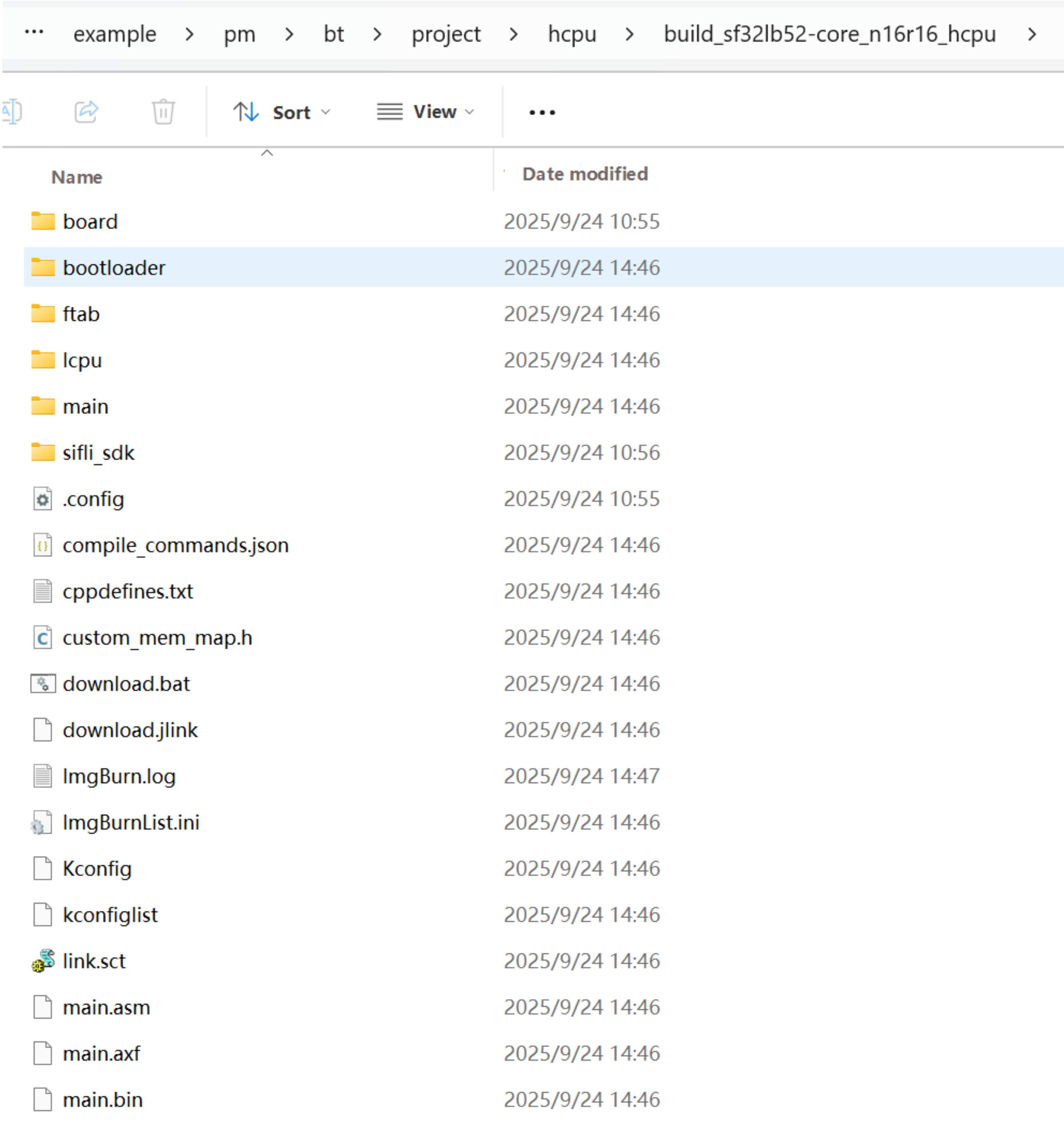
3.2.1. Compilation

If using pre-compiled image files, you can skip directly to the programming section to start testing.

Enter the example\pm\bt\project\hcpu directory and execute

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

Compile and generate HCPU image file (no need to compile LCPU project separately, LCPU will be automatically compiled and LCPU bin will be packaged into HCPU image when compiling HCPU). The compiled image file is saved in the build directory.



3.2.2. Programming Image

Execute

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

in the command line compilation directory to program the image file generated by compilation in the build directory.

3.2.3. Transmit Power Configuration

The project is configured with a default transmit power of 0 dBm. Use the command ble_tx_pwr_save x to adjust transmit power, where x represents the target power value in dBm.Example: To set transmit power to 10 dBm, run ble_tx_pwr_save 10.The device will automatically reboot for the setting to take effect after modification.

3.3. Device Initialization

When using the development board to test Bluetooth for the first time, the following initialization commands need to be executed after power-on (after programming the file system image, the Bluetooth address will be lost and needs to be reconfigured). Commands are executed in the HCPU console, and the wake-up PIN needs to be connected to low level to prevent HPSYS from entering low power mode, otherwise HCPU cannot process commands.

- nvds reset_all 1
- nvds update addr 6

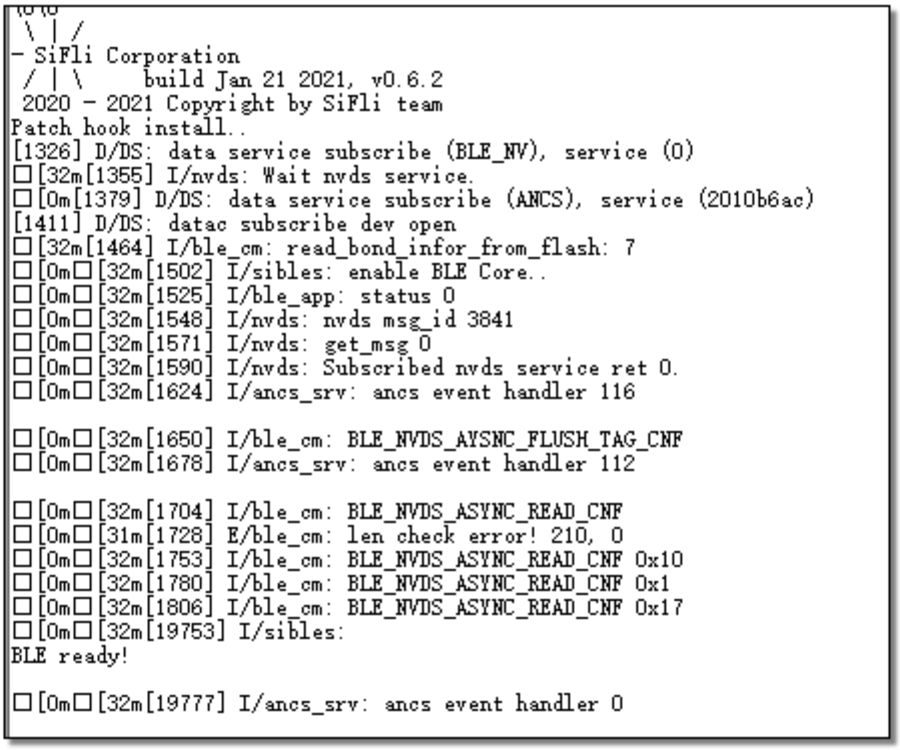
For example: nvds update addr 6 1234567890C8. Note that the Bluetooth address format needs to be maintained in a certain format. It is recommended not to change C8, and users can change other parts. After command execution is completed, press the Reset key to restart the device.

3.4. ADV Scenario

Note: It is recommended to conduct ADV scenario tests in a shielding box or shielding room. Otherwise, interference from other devices will lead to inflated test results.

1. ◦ Open serial debugging tool, connect to HCPU console serial port, connect measurement device to the module under test

2. ◦ Connect wake-up PIN to low level, press Reset key on the baseboard to reset. After successful startup, the log shown in the figure below appears



```
 \ | /
- 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: datac 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
```

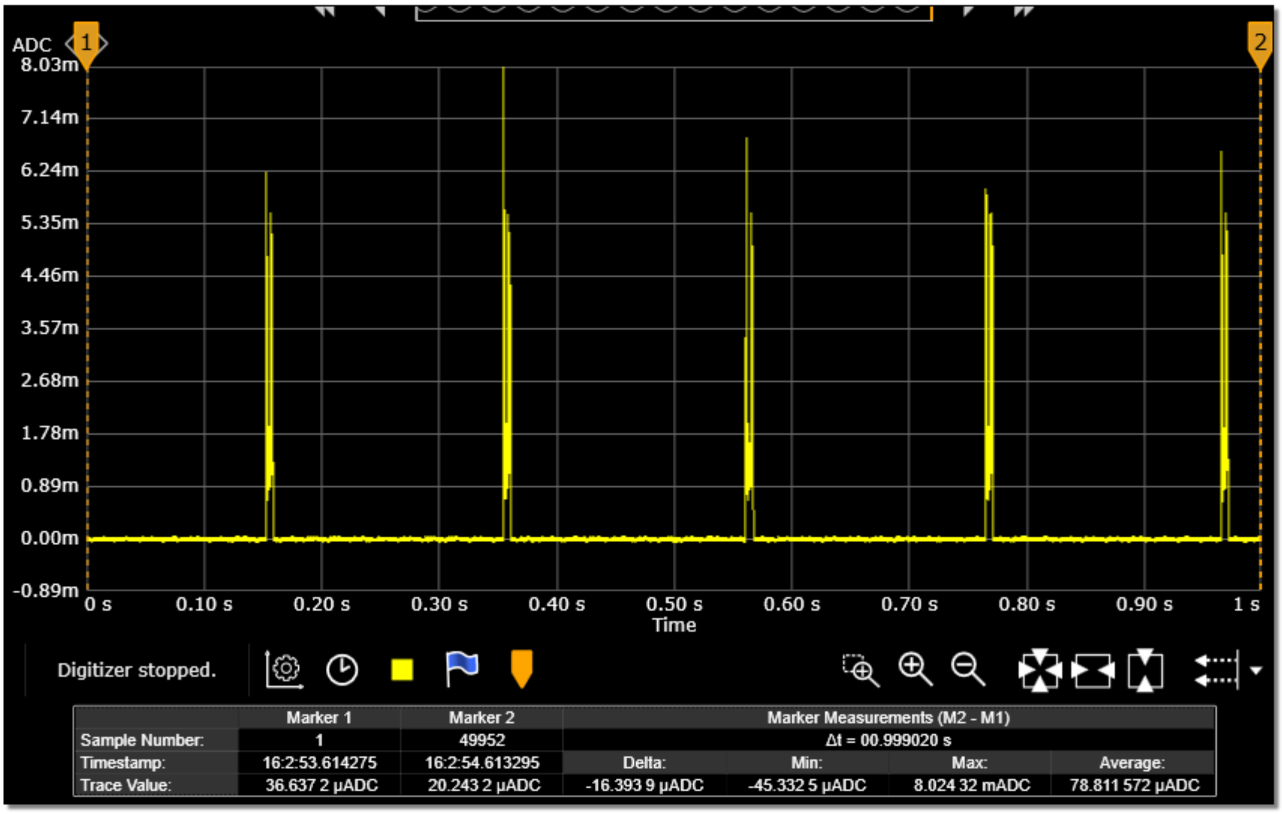
3. ◦ The default ADV period after startup is 200ms. Since Inquiry Scan and Page Scan will also be automatically enabled, to test BLE power consumption, you need to use btskey command to disable Scan. Refer to the Scan section of classic Bluetooth for disable commands. For example, first send btskey s command. If it shows you are currently in the main menu, you can send the following three commands in sequence to disable Page Scan and Inquiry Scan. After sending btskey0, you can send btskey 4 again to query the Scan status.

```
(a) btskey 1
(b) btskey 7
(c) btskey 0
```

1. ◦ Connect wake-up PIN to high level, system enters low power mode, current value drops significantly as shown in the figure below. Measure current at 200ms interval. The current waveform in low power mode is shown in the ADV=200ms current waveform diagram. Record the 10-second average current

as C1, the current between two peaks as sleep current C2, and the ADV incremental current C=C1-C2

Current change when entering low power mode



Current waveform for ADV=200ms

1. ◦ Connect wake-up PIN to low level, system exits low power mode, send command ble_config adv 500 in console to change ADV interval to 500ms

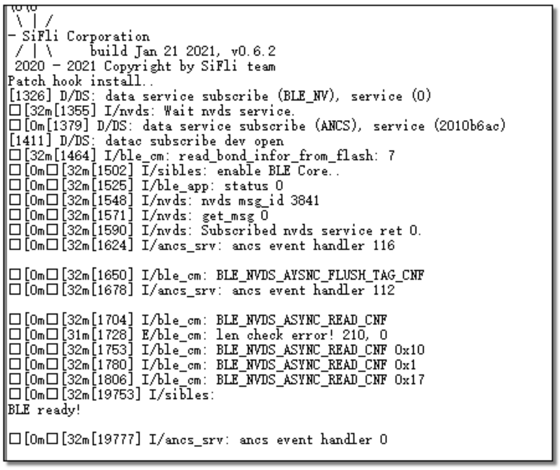
2. ◦ Connect wake-up PIN to high level, system enters low power mode again, measure current at 500ms interval

Repeat steps 5 and 6 to measure current when ADV interval is 1000ms.

3.5. Connection Scenario

1. ◦ Open serial debugging tool, connect to HCPU console serial port, connect measurement device to the module under test

2. ◦ Connect wake-up PIN to low level, press Reset key on the baseboard to reset. After successful startup, the log shown below appears. Similar to measuring ADV current in the ADV chapter, BT Scan also needs to be disabled



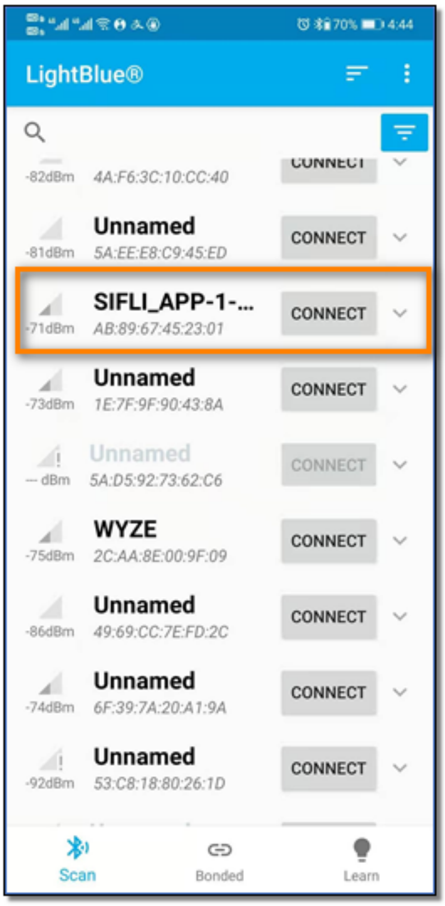
```
 \ | /
- SiFli Corporation
 / | \     build Jan 21 2021, v0.6.2
 2020 - 2021 Copyright by SiFli team
Patch hook install..
[1326] D/DS: data service subscribe (BLE_NV), service (0)
□[32m[1355] I/nvds: Wait nvds service.
□[0m[1379] D/DS: data service subscribe (ANCS), service (2010b6ac)
[1411] D/DS: datac 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
```

1. ◦ Open the LightBlue software on your phone as shown below, find the device named SIFLI_APP in the Scan list, click CONNECT to connect to the device



1. ◦ After successful connection, BLE enters connection state, with default connection period of 15ms
2. ◦ Connect wake-up PIN to high level, system enters low power mode, measure current at 15ms interval
3. ◦ Connect wake-up PIN to low level, system exits low power mode, send command ble_config conn 50 in console to change connection period to 50ms. Confirm that console shows the print as shown below, where "Updated connection interval: 40" indicates connection interval in 1.25ms units, 40×1.25=50ms. If no print appears, it means parameter update failed and the command needs to be sent again. Also observe current waveform to confirm whether interval update is successful

```
msh >
msh >ble_config conn 50
□□□\0Q\0X缨□a□□ □□\0(\0(\0\0\0?□\0\0□□□□\0R\0晒?b□□□\0□□ msh >
msh >□[32m[2017749] I/sibles: GAP event 3584
□[0m[2017772] D/BLE_GAP: GAPC event status, op: 9, ret 0
□[32m[2017798] I/ancs_srv: ancs event handler 63

□[0m□[32m[2017827] I/sibles: 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. ◦ Connect wake-up PIN to high level, system enters low power mode again, measure current at 50ms period. Similar to ADV current, measure 10-second average current C1, record current between two peaks as sleep current C2, and the incremental current in connection state is C=C1-C2

Repeat steps 6 and 7 to measure current when connection period is 200ms, 500ms and 1000ms.

3.6. Measurement Results Summary

The current values for ADV and Connection in the table are incremental currents for those modes. Sleep is the sleep current. The actual average current equals the sum of incremental current and sleep current. For example, ADV at 1-second interval with 4dBm transmission power has an average current of 10.5+14.5=24.5uA

3.6.1. BLE Power Consumption Table

Mode	Condition	Power Supply Voltage 3.3V @TXpower=0dBm	Power Supply Voltage 3.3V @TXpower=4dBm	Power Supply Voltage 3.3V @TXpower=10dBm	Power Supply Voltage 3.3V @TXpower=13dBm	Uni
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

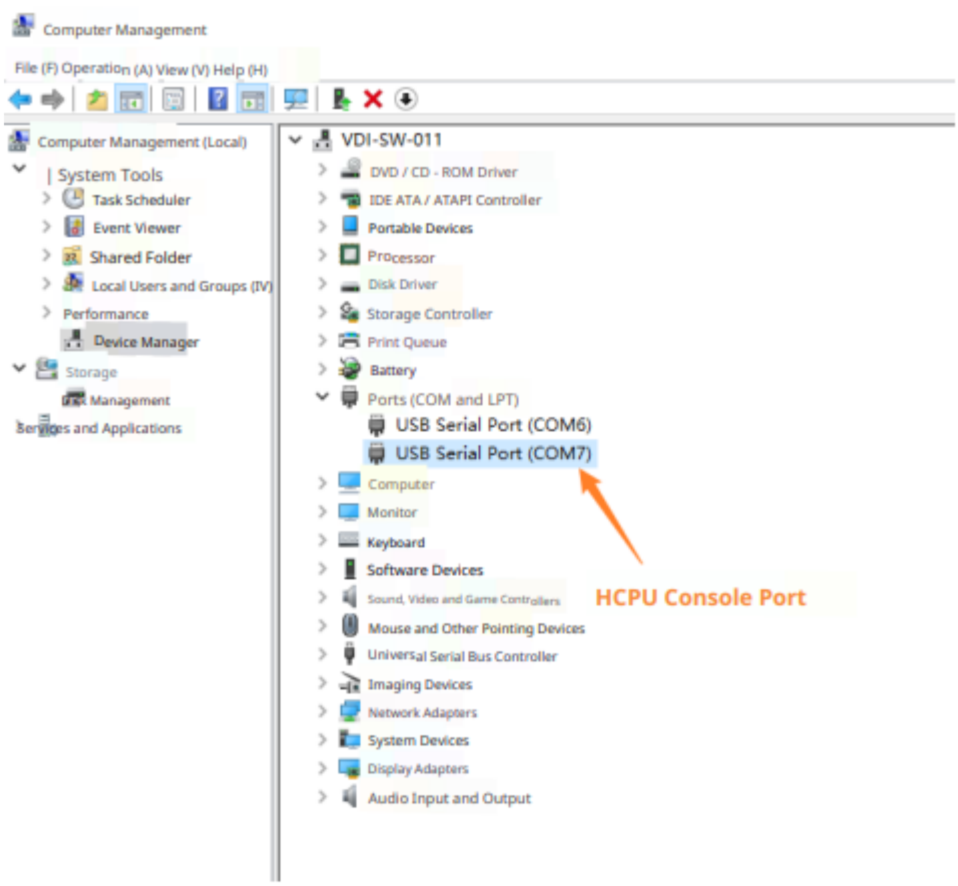
4. Classic Bluetooth Power Consumption Test

Classic Bluetooth provides higher data transmission rates than BLE, suitable for audio transmission, data synchronization and other application scenarios. This chapter provides detailed introduction to power consumption characteristics of classic Bluetooth in various working modes.

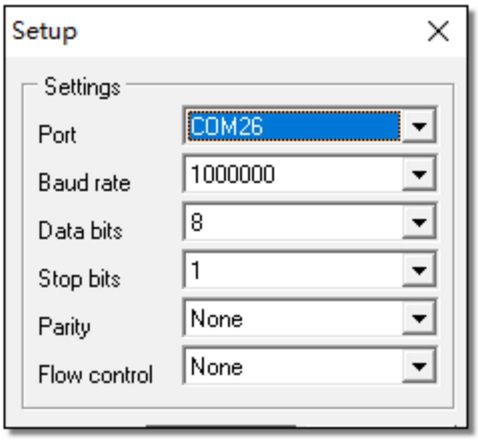
4.1. Test Description

Use BT routine to test power consumption of Scan and Sniff modes. After system power-on, the test routine automatically enables Scan and ADV. Use phone to connect to Bluetooth device. Commands can be sent in HCPU console to modify configuration. All commands must end with carriage return and line feed.

After connecting PC and debug board with USB Type-C cable, two serial ports will be enumerated, where HCPU uses the second serial port as console port, as shown below.



Serial port settings are shown below, with baud rate set to 1000000.



For convenient control of test conditions, use PA24 as HCPU wake-up PIN. When wake-up PIN is at low level, HCPU cannot enter low power mode. At this time, commands can be sent to HCPU through console to modify parameters. When PA24 is floating or connected to high level (i.e., 3.3V voltage, high level refers to 3.3V voltage unless otherwise specified below), HCPU enters low power mode, and LCPU periodically enters and exits low power mode. At this time, console cannot be used.

In the test routine, LCPU main frequency is 24MHz, HCPU low power mode is Deepsleep, and LCPU low power mode is Standby. HCPU uses btskey command to operate menu to modify configuration.

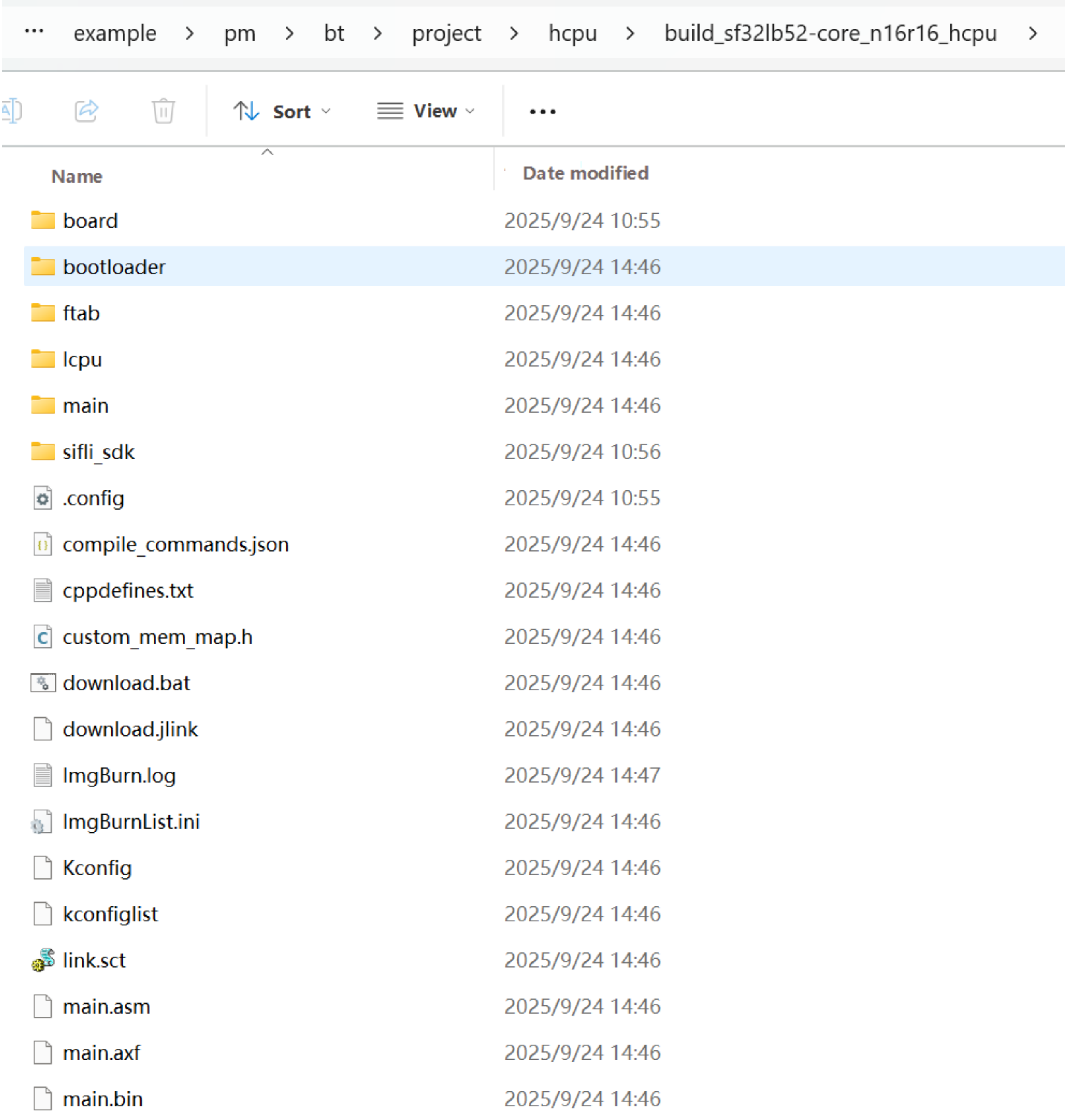
4.2. Routine Compilation and Programming

4.2.1. Compilation

Enter the example\pm\bt\project\hcpu directory and execute

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

Compile and generate HCPU image file. The compiled image file is saved in the build directory.



4.2.2. Programming Image

Execute in the command line compilation directory

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

to program the image file generated by compilation in the build directory.

4.2.3. Transmit Power Configuration

The project is configured with a default transmit power of 0 dBm. Use the command ble_tx_pwr_save x to adjust transmit power, where x represents the target power value in dBm.Example: To set transmit power to 10 dBm, run ble_tx_pwr_save 10.The device will automatically reboot for the setting to take effect after modification.

4.3. Device Initialization

When using the development board to test Bluetooth for the first time, the following initialization commands need to be executed after power-on. Commands are executed in HCPU console, and wake-up PIN needs to be connected to low level to prevent HPSYS from entering low power mode, otherwise HCPU cannot process commands.

1. nvds reset_all 1

2. nvds update addr 6

For example: nvds update addr 6 1234567890C8. Note that the Bluetooth address format needs to be maintained in a certain format. It is recommended not to change C8, and users can change other parts. After command execution is completed, press the Reset key to restart the device.

4.4. Scan

1. ◦ Open serial debugging tool, connect to HCPU console serial port, connect measurement device to the module under test

2. ◦ Connect wake-up PIN to low level, reset development board, after successful startup the log shown below appears

```
 \ | /
- 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 sucess

[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.audprc main: init 00 ADC_PATH_CFG0 0x924

[134] I/drv.audprc main: HAL_AUDPRC_Init res 0

[136] I/drv.audcodec main: HAL_AUDCODEC_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 2046
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 addreess: 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
```

1. ◦ After startup, ADV, Inquiry Scan and Page Scan will all be enabled. To test Scan power consumption, ADV needs to be disabled first. The disable command is

```
diss adv_stop
```

1. ◦ After boot, the system is in BTS main menu by default. You can configure to enable or disable Scan by sending btskey commands. Send btskey s command to display current menu, then send commands according to menu prompts to enter submenus. For example, in main menu, sending the following three commands in sequence can enable Page Scan and disable 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. SPP Client                                   ##
##    3. SPP 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                          ##
##                                                    ##
########################################################
```

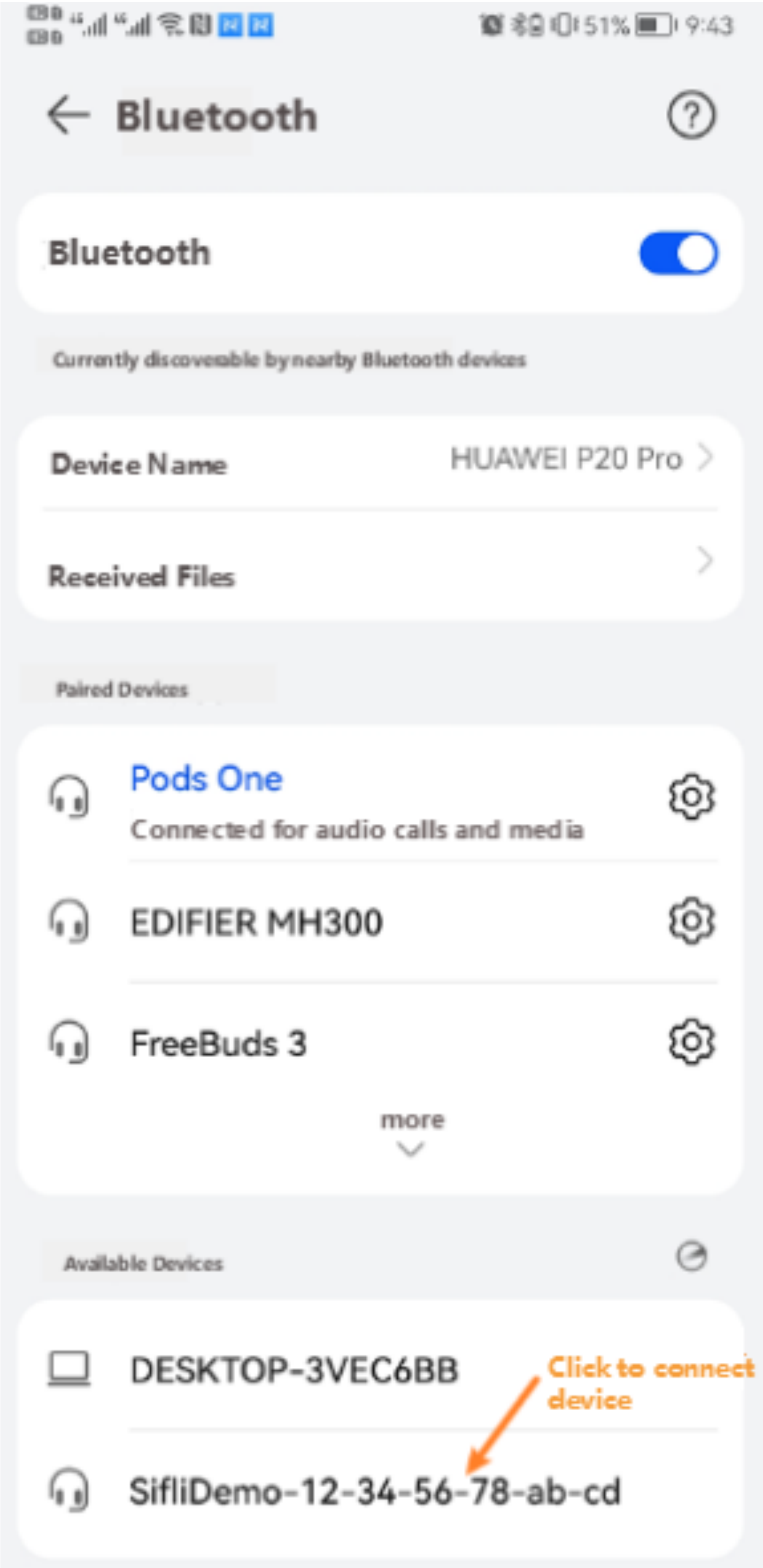
1. ◦ Send BTS commands to configure device to send only Inquiry Scan or Page Scan, connect wake-up PIN to high level, system enters low power mode, measure 1-minute average current, record as Scan average current C1, also measure the base current between two peaks, record as sleep current C2, Scan incremental current is C=C1-C2. The Page Scan period in test program is 1.28 seconds, Inquiry Scan period is 2.56 seconds, so Inquiry Scan incremental current is half of Page Scan.

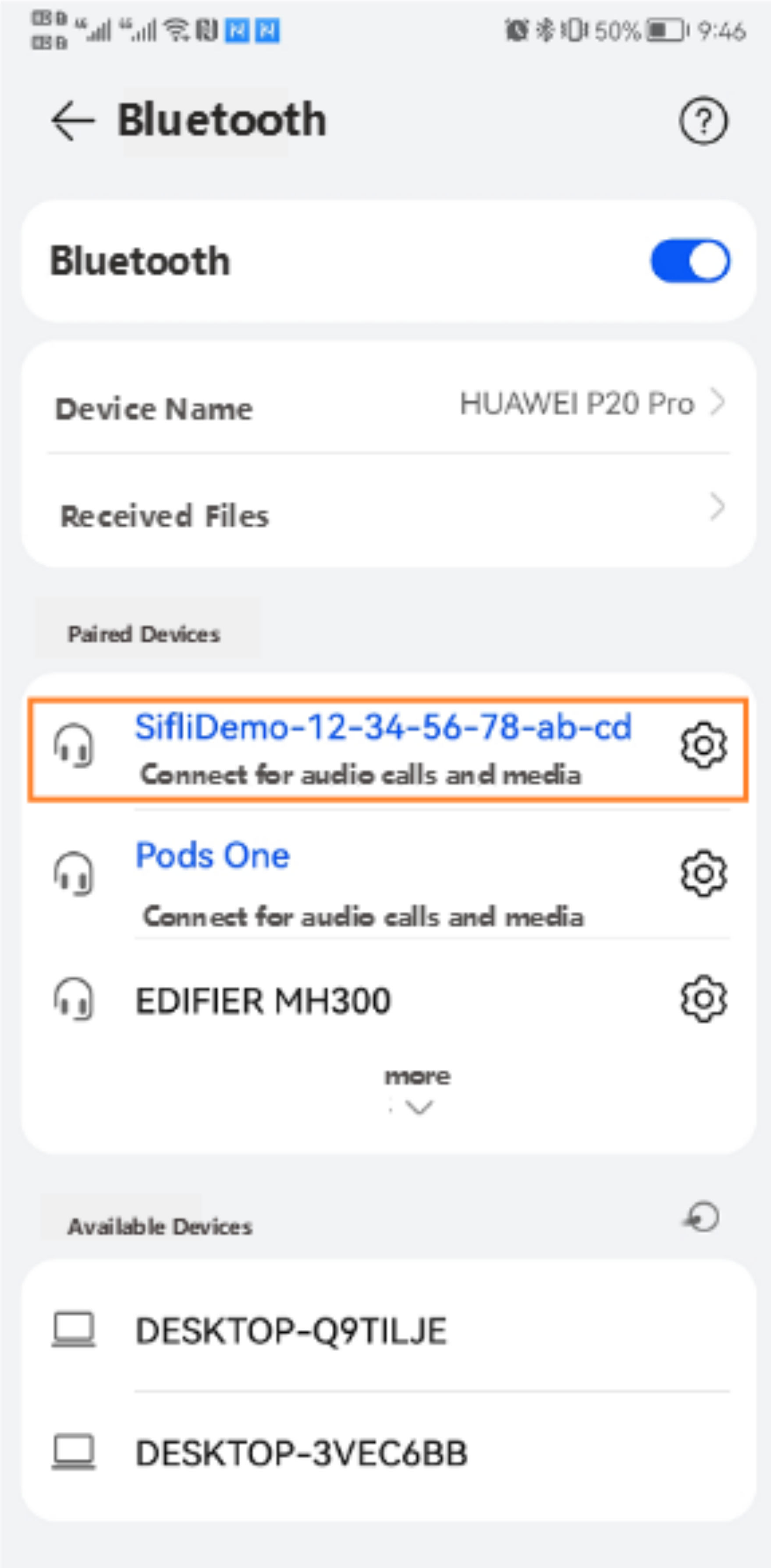
2. ◦ Connect wake-up PIN to low level, send BTS commands to configure device to send both Inquiry Scan and Page Scan simultaneously, then connect wake-up PIN to high level, system enters low power mode, measure test average current, record as Both Scan current

4.5. Sniff Mode

1. ◦ Refer to the Scan chapter steps to reset development board and disable ADV

2. ◦ Phone connects to the corresponding Bluetooth device on development board, click to bring up pairing window, click confirm to pair successfully, Bluetooth device enters paired device list and shows connected status, as shown below:


9:43
Bluetooth
Bluetooth
Currently discoverable by nearby Bluetooth devices
Device Name
HUAWEI P20 Pro
Received Files
Paired Devices
Pods One
Connected for audio calls and media
EDIFIER MH300
FreeBuds 3
more
Available Devices
DESKTOP-3VEC6BB
Click to connect device
SifliDemo-12-34-56-78-ab-cd


Bluetooth
Bluetooth
Device Name
HUAWEI P20 Pro
Received Files
Paired Devices
SifliDemo-12-34-56-78-ab-cd
Connect for audio calls and media
Pods One
Connect for audio calls and media
EDIFIER MH300
more
Available Devices
DESKTOP-Q9TILJE
DESKTOP-3VEC6BB

1. ◦ After connection is completed, wait for a while, console window will print »Sniff mode, indicating device has entered Sniff mode

2. ◦ Send btskey commands to disable Scan, specific steps are as follows:

```
(a) btskey s: View current menu
(b) If it's HFP HP Menu, then send
(c) btskey r: Return to previous menu BTS2 Demo Main Menu, then send
following commands in sequence
(d) btskey 1: Select Generic Command
(e) btskey 7: Select Scan mode
(f) btskey 0: Disable scan
```

1. ◦ Similar to Scan current measurement method, Sniff mode incremental current can be obtained by subtracting sleep current between two peaks from 10-second average current

4.6. Measurement Results Summary

The current values for Sniff mode and Scan in the table are incremental currents for those modes. Sleep is the sleep current. The actual average current equals the sum of incremental current and sleep current. For example, in Sniff mode, 500ms interval, attempt=4, 10dBm transmission power, average current is 23+16.6=39.6uA

4.6.1. BT Power Consumption

Mode	Condition	Power Supply Voltage 3.3V @TXpower=0dBm	Power Supply Voltage 3.3V @TXpower=4dBm	Power Supply Voltage 3.3V @TXpower=10dBm	Power Supply Voltage 3.3V @TXpower=13dBm	Unit
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

Mode	Condition	Power Supply Voltage 3.3V @TXpower=0dBm	Power Supply Voltage 3.3V @TXpower=4dBm	Power Supply Voltage 3.3V @TXpower=10dBm	Power Supply Voltage 3.3V @TXpower=13dBm	Unit
Sleep	16.6	16.6	16.6	16.6	16.6	uA

1. Scan receives 11.25ms every 1.28s, Both Scan receives 22.5ms every 1.28s.

2. bt 500ms sniff@TXpower10dBm: =23+16.6=39.6uA

5. Processor Performance and Power Consumption Test

This chapter focuses on power consumption characteristics analysis of SF32LB52x processor under different computational loads, including standardized benchmark tests and basic loop tests.

5.1. Test Objectives

Evaluate chip power consumption performance under different computational intensities through standardized processor tests:

- Coremark Benchmark Test: Industry-standard CPU performance benchmark test
- While Loop Test: Power consumption baseline test for basic loop operations
- Shutdown Mode Test: Power consumption measurement in the lowest power state

5.2. Test Description

CoreMark routine tests power consumption in the following scenarios:

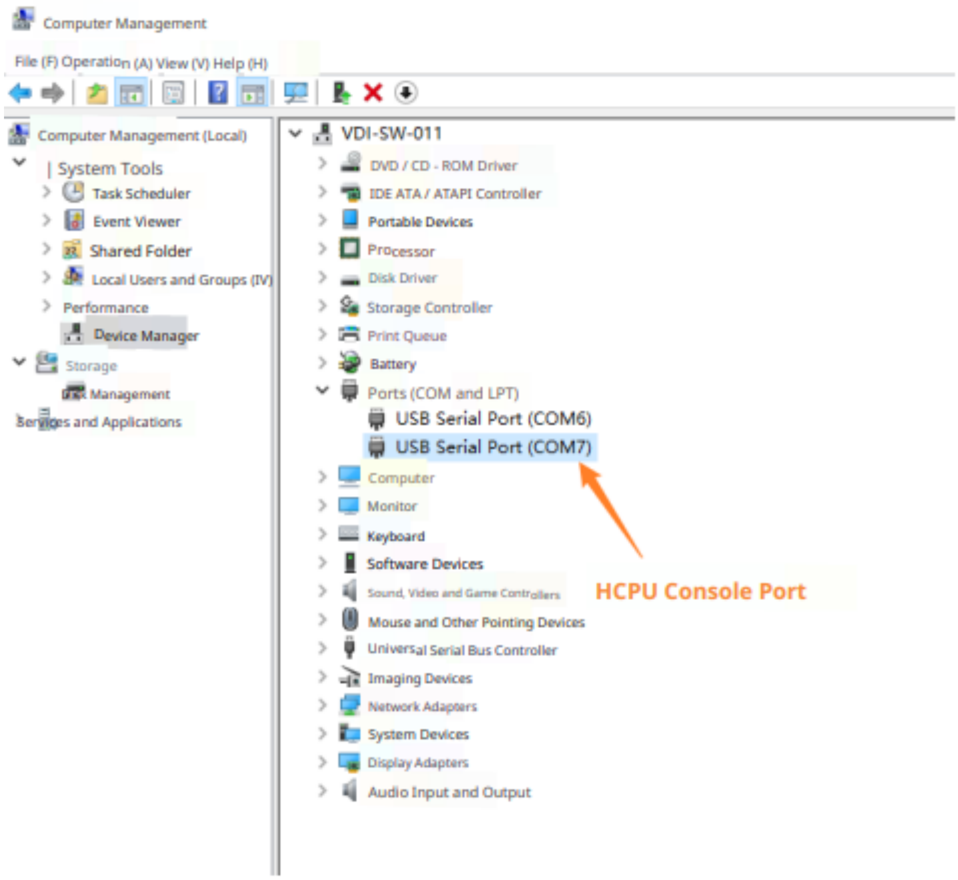
- One core executes CoreMark benchmark test program
- One core executes while loop for a period of time, executing nop instruction in the loop
- System shutdown, can be woken up by RTC timer
- System shutdown, woken up by key press

When LCPU executes test commands, HCPU automatically enters DeepSleep low power mode. During this period, HCPU cannot respond to console commands. After test completion, HCPU exits low power mode and new test commands can continue to be sent. If LCPU test never ends, you can connect PA30 to low level to force wake up HCPU. However, note that when measuring current, PA24 cannot be connected to low level, it must be connected to high level or left floating, otherwise it will cause measurement values to be high.

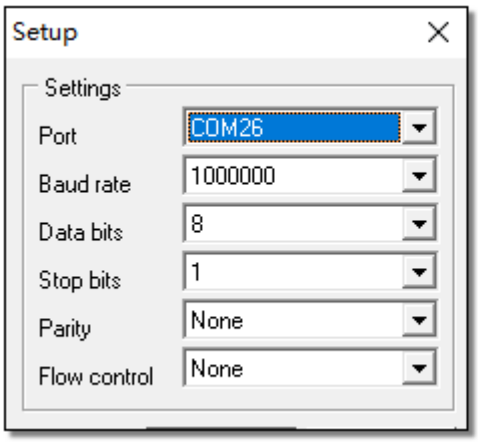
LCPU never enters low power mode. When not executing tasks, LCPU is in WFI state.

Note that all commands sent must end with carriage return and line feed characters.

After connecting PC and debug board with USB Type-C cable, two serial ports will be enumerated. HCPU uses the second serial port as console port, as shown below.



Serial port settings are shown below, with baud rate set to 1000000.



HCPU can use the following commands to start corresponding tasks:

- run_coremark freq_in_mhz: Modify main frequency and execute CoreMark, freq_in_mhz is frequency in MHz units For example: run_coremark 48, execute CoreMark at 48MHz main frequency

- run_while_loop freq_in_mhz: Modify main frequency and execute while loop for a period of time, freq_in_mhz is frequency in MHz units For example: run_while_loop 48, execute while loop at 48MHz main frequency

Use HCPU console to send the following commands to make LCPU execute specified tasks:

- tolcpu run_coremark freq_in_mhz
- tolcpu run_while_loop freq_in_mhz

Command meanings are the same as HCPU. HCPU also supports the following command:

- shutdown [wakeup_time_in_sec]: Shutdown, wakeup_time_in_sec is optional parameter in seconds, indicating how long after shutdown to automatically power on. If no parameter is provided, it can only be woken up by KEY1 key press after shutdown.

Note: Due to compilation using Ofast optimization level, benchmark results are for reference only and cannot reach the highest score when using Omax.

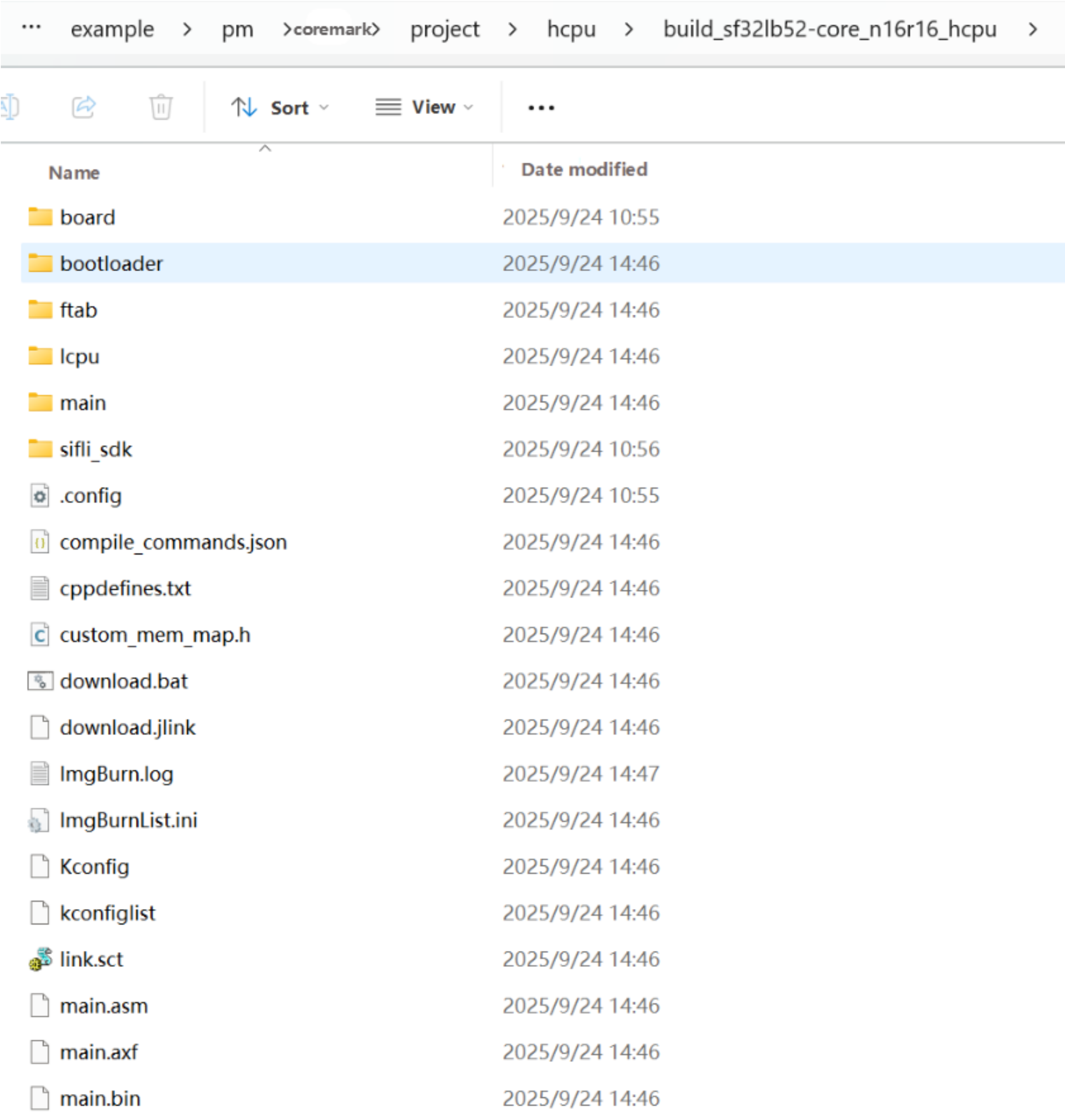
5.3. Routine Compilation and Programming

5.3.1. Compilation

If using pre-compiled image files, you can skip directly to the programming section to start testing. Enter the example\pm\coremark\project\hcpu directory and execute

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

Compile and generate HCPU image file. The compiled image file is saved in the build directory.



5.3.2. Programming Image

Execute in the command line compilation directory

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

to program the image file generated by compilation in the build directory.

5.4. CoreMark

5.4.1. HPSYS

1. ◦ Open the serial debugging tool, connect the HCPU console serial port, and connect the measurement device to the module under test

2. ◦ Reset, after successful startup, the following log appears in the HCPU console as shown in the figure below

```
\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. ◦ Send command run_coremark 192 to get average current C1, send run_coremark 168 to get average current C2, obtain the incremental current for 192MHz and 168MHz: C=(C1-C2)/(192-168)

2. ◦ Send command run_coremark 144 to get average current C1, send run_coremark 120 to get average current C2, obtain the incremental current for 144MHz and 120MHz: C=(C1-C2)/(144-120)

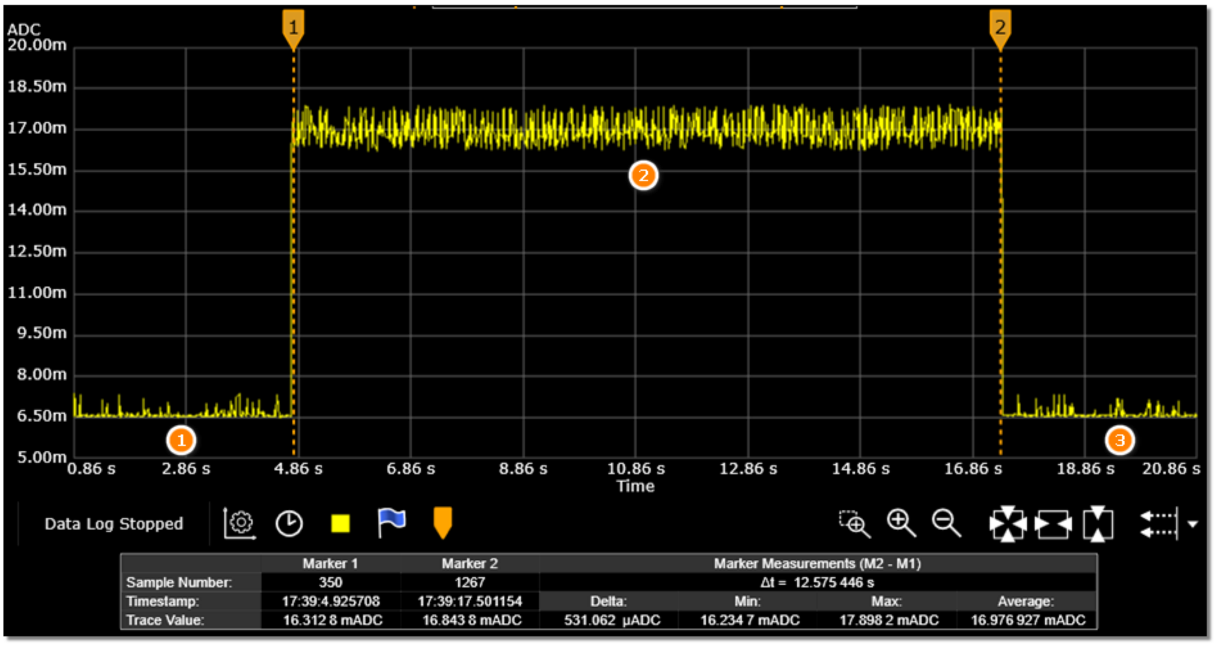
3. ◦ Send command run_coremark 48 to get average current C1, send run_coremark 24 to get average current C2, obtain the incremental current for 48MHz: C=(C1-C2)/(48-24)

4. ◦ Send command run_coremark 24 to get average current C1, send run_coremark 12 to get average current C2, obtain the incremental current for 24MHz and 12MHz: C=(C1-C2)/(24-12)

- Note: The 24 MHz clock has two modes: D0 and D1. D0 is used by default. To test the D1 mode, append parameter 1 to the command: run_coremark 24 1.

1. ◦ As shown in the figure below, Phase 1 is the current waveform in WFI mode when HCPU runs at 192MHz main frequency. After starting to execute CoreMark, it enters Phase 2, where the current rises

and remains until the test ends. Phase 3 is the current waveform returning to WFI mode



For frequencies such as 48MHz/24MHz/12MHz, the coremark execution time will be very long. It is not necessary to wait for the test to complete. After measuring the current value, you can reset and test the next item.

5.5. While Loop

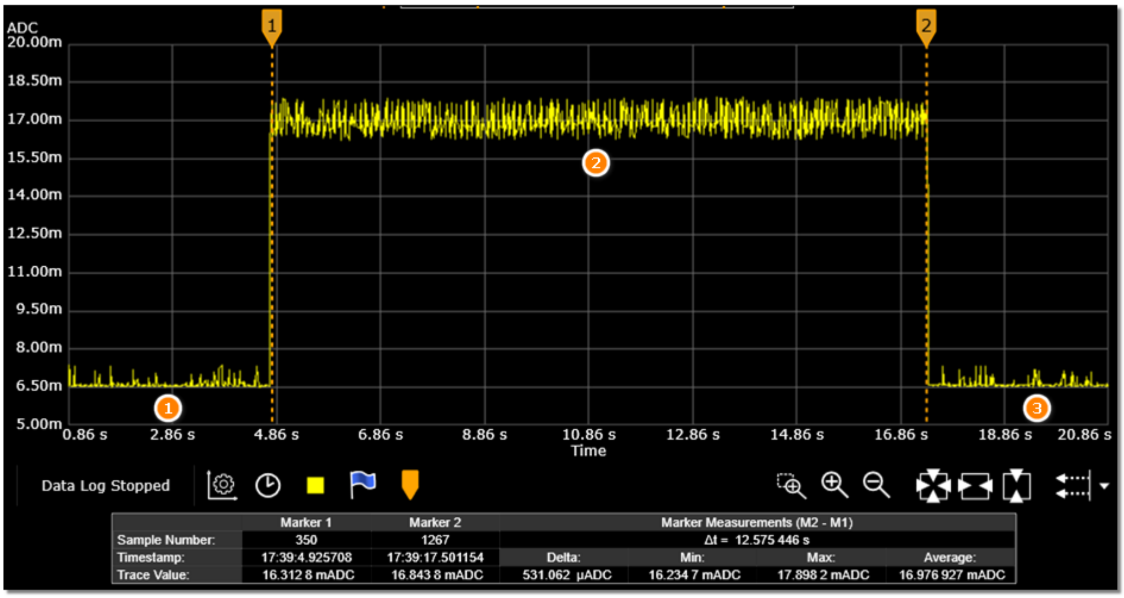
5.5.1. HPSYS

1. ◦ Open serial debugging tool, connect to HCPU console serial port, connect measurement device to the module under test

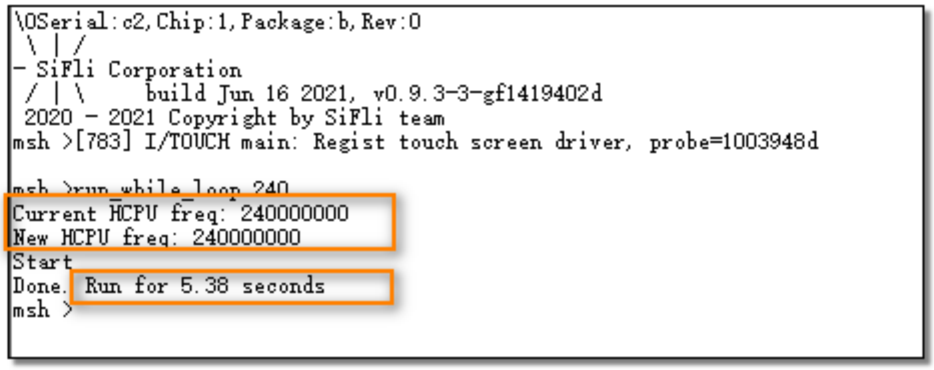
2. ◦ Reset, after successful startup the log shown below appears in HCPU console

```
\0Serial: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. ◦ Send command run_while_loop 192, instructing HCPU to execute while loop test at 192MHz frequency, measure average current C1 at this time. As shown in the HCPU While Loop test current diagram below, phase 1 is the current waveform in WFI mode when HCPU runs at 192MHz main frequency. After starting while loop execution, it enters phase 2, current rises and maintains until test completion. Phase 3 is the current waveform returning to WFI mode. Console displays HCPU frequency and while loop duration, as shown in the HCPU While Loop Log diagram.



HCPU While Loop Test Current Diagram



HCPU While Loop Log Diagram

1. ◦ Send command run_while_loop 168, instructing HCPU to execute while loop test at 168MHz, measure average current C2 at this time, thus obtaining current increment per MHz as C=(C1-C2)/(192-168)

2. ◦ Execute run_while_loop 144 and run_while_loop 120, measure average current at 144MHz and 120MHz, obtain incremental current at 144MHz as C=(C1-C2)/(144-120)

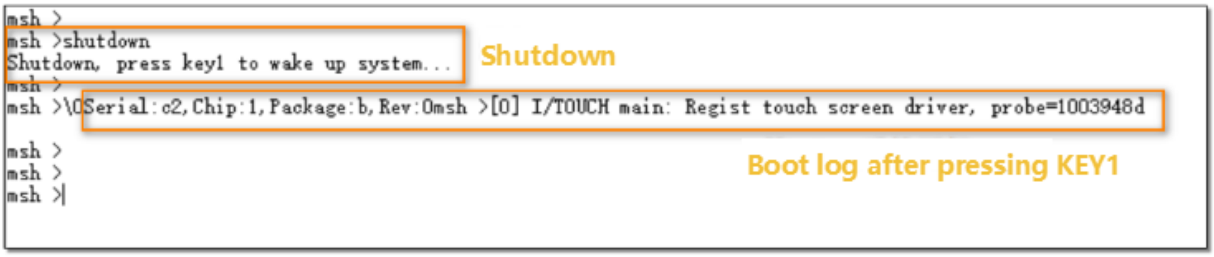
3. ◦ Execute run_while_loop 48 and run_while_loop 24, measure average current at 48MHz and 24MHz, obtain incremental current at 48MHz as C=(C1-C2)/(48-24)

4. ◦ Execute run_while_loop 24 and run_while_loop 12, measure average current at 24MHz and 12MHz, obtain incremental current at 24MHz as C=(C1-C2)/(24-12)

5.6. Shutdown

1. ◦ Open serial debugging tool, connect to HCPU console serial port

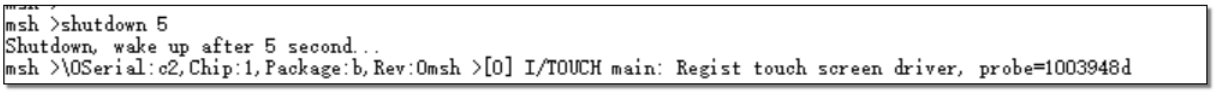
2. ◦ Reset, after successful startup the log shown below appears in HCPU console



3. ◦ Send command shutdown, instructing system to shut down. After shutdown, it can only be woken up by KEY1 key press. Since shutdown current is relatively small, it is recommended to use a multimeter for measurement

4. ◦ Press KEY1 key to wake up system, console shows startup phase log again, but version number log will not appear

5. ◦ Send command shutdown 5, as shown below, instructing system to shut down and automatically power on after 5 seconds. For convenient measurement, the waiting time can be extended



5.7. Measurement Results Summary

5.7.1. Processor Power Consumption

5.7.1.1. CoreMark Test Results

Clock Freq	3.3V Power Supply Current (uA)	Delta (uA/MHz)	CoreMark
240MHz	9380	35	240MHz
216MHz	8560		216MHz
192MHz	7760		192MHz
168MHz	6920		168MHz
144MHz	5350	27	144MHz

Clock Freq	3.3V Power Supply Current (uA)	Delta (uA/MHz)	CoreMark
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 Test Results

Clock Freq	3.3V Power Supply Current (uA)	Delta (uA/MHz)
240MHz	6910	24
216MHz	6340	
192MHz	5790	
168MHz	5200	
144MHz	4080	25
120MHz	3470	
48MHz	1400	17
24MHz D0	806	27

Clock Freq	3.3V Power Supply Current (uA)	Delta (uA/MHz)
12MHz	482	
24MHz D1	993	•

5.7.1.3. Shutdown Power Consumption

Test Item	3.3V Power Supply Current (uA)
Key Wakeup	1.3