

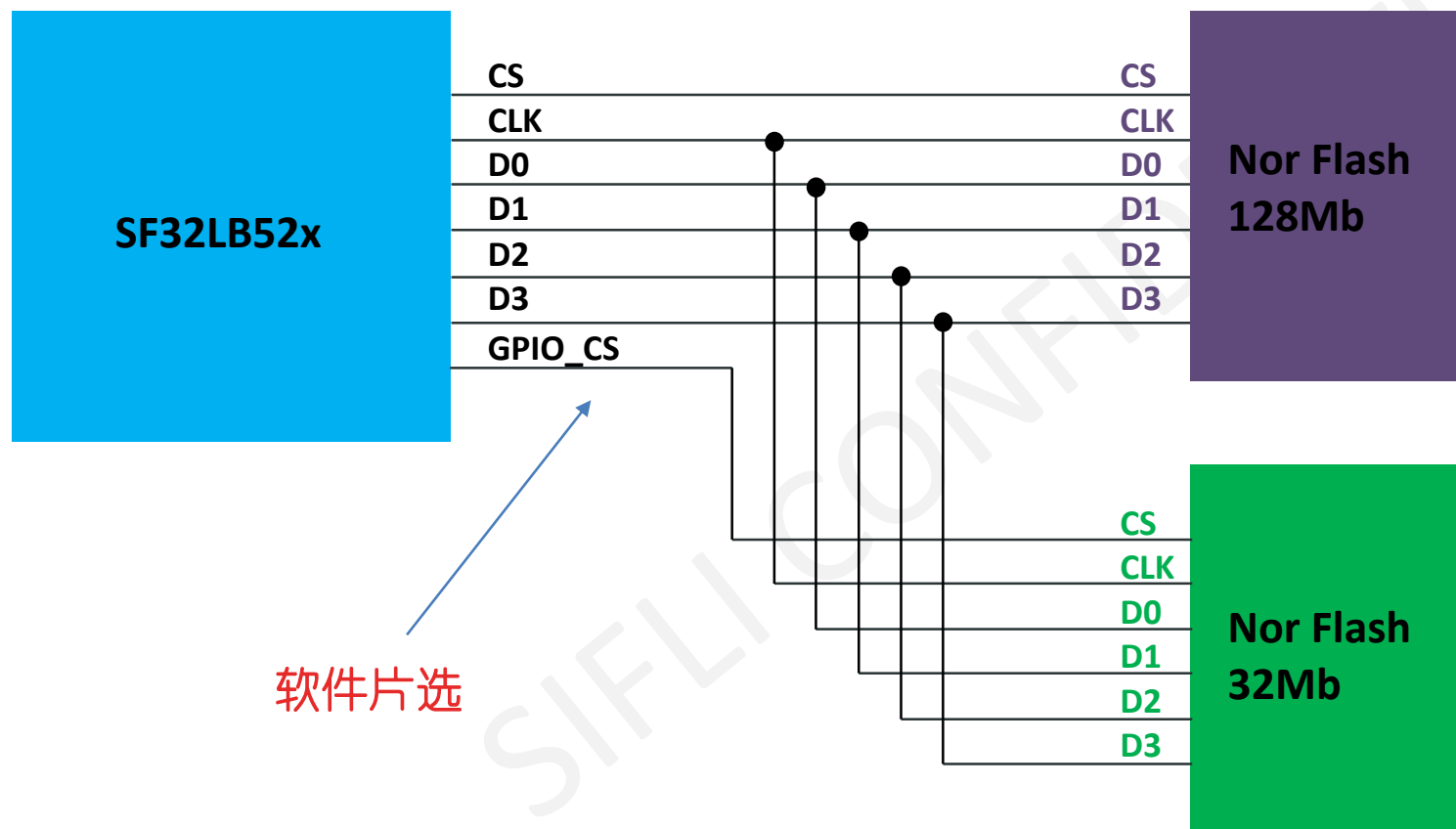


SiFli 52x 支持双Flash 方案

双Flash方案介绍

- 双Flash硬件框图
- 双Flash软件方案
- 双Flash空间分配

双Flash硬件框图



A区起始地址: 0x10000000

B区起始地址: 0x18000000

双flash的方案已经在小米项目上验证。

考虑到Flash的切换需要软件片选，因此切换需要CPU参与。为了使方案稳定可靠，对两个Flash的存放内容需要进行换分，划分原则如下：（以下介绍中把Flash 1的存储空间称为A区，把Flash 2存储空间称为B区。）

- A区：不受限制。支持XIP、硬件访问（EZIP/EPIC/DMA）、直接访问。

存放主代码（code）、图片、字体（直接访问）

- B区：不支持XIP、不支持硬件直接寻址访问（EZIP/EPIC/DMA）。只支持调用驱动进行访问。

OTA的升级区域（pong）、OTA manager、表盘压缩包、字体（文件系统访问）、死机Dump、NV（包括历史）、打点



- 访问A区的起始地址是0x12000000，B区的其实地址是0x13000000
- B区访问**只能通过CPU调用驱动**的方式实现。
- 上层软件对A、B区是透明的，驱动将自动识别A、B区调用原子操作实现擦、写、读实现读写的功能。

（如果需要启用后台擦B区的功能，需要软件显式设置）。



表盘：

■ 表盘传输过程

user bin接收表盘传输过程中，边接收表盘压缩包，边调用驱动写入B区的表盘区域

■ 表盘装载过程

表盘被选中后，调用驱动从B区的表盘区域搬移资源到PSRAM。

■ 表盘运行

表盘直接访问PSRAM资源运行。

OTA：

■ OTA传输过程

user bin边接收OTA.BIN，边调用驱动写入B区的OTA区域

■ OTA解压覆盖

user bin接收完成后，进入boot loader完成解压。此时，读B区的OTA.bin采用驱动进行读取。

KVDB、TSDB、log、打点

■ 通过flash的驱动访问，与之前一样

字体、多语言

■ 通过文件系统进行访问。用户无感知

产线烧写调用驱动

■ 工具与之前一样



双Flash的空间分配

双NOR Flash 分配建议			
	模块	大小/KB	详细说明
	ftab	16	
A区	bootloader	48	
	ota manager	384	这个可以和code. bin合并，用ramrun来实现。
	code		
	图片资源		
	字体		—(直接访问)—
	Total	16384	
B区	KVDB/TSDB		
	OTA缓存区域		
	字体、多语言		文件系统访问
	铃声		
	表盘		
	用户自定义的数据		
	Total	8192	

思澈科技



人工智能物联网时代的MCU新秀

附：字体/多语言使用文件系统的配置

C:\Windows\System32\cmd.exe

(Top) → Freetype and bitmap support

Sifli Configuration

[*] OTA err font built in support

[*] Nor solution font using file_system(for dual_flash)

[] Freetype any size support(from 12 -> 36)

[*] Bitmap is used

FLASH2	OFFSET (HEX)	SIZE (KB)	PARTITION TYPE	XIP	IS RESET	GC_RATIO	FS_DIR
upay	10000	16	RSV		Y		
upay_safety	14000	8	RSV		N		
dfu	16000	16	RSV		Y		
ble	1A000	16	RSV		Y		
app	1E000	440	RSV		Y		
msg	8C000	0	RSV		Y		
contact	8C000	16	RSV		Y		
his_hr	90000	16	RSV		Y		
his_sport	94000	16	RSV		Y		
his_sleep	98000	16	RSV		Y		
his_call_log	9C000	16	RSV		Y		
dfu_mng_code	A0000	512	BIN	120A0000			DFU
solution_code	120000	3312	BIN				HCP
img	45C000	8576	BIN				HCP
font	CBC000	1984	BIN				HCP
tiny_font	5AC000	0	BIN				HCP
image-ota	5AC000	0	BIN				HCP
dfu_download_buffer	5AC000	0	BIN				DFU
root	EAC000	1360	BIN			9 /	HCP
dyn	10000000	0	BIN				/dyn

语言名称

多语言选择 >> 右键批量

en_us(English_US)	内置
en_GB(English_GB)	外置
zh_cn(简体中文)	外置
zh_HK(繁体中文)	外置
ru_RU(русский)	外置
de_DE(Deutsch)	外置
it_IT(Italiano)	外置
fr_FR(Français)	外置
pl_PL(Polski)	外置
pt_PT_EURO(Português)	外置
es_ES_EURO(Idioma esp...	外置
Hi_IN(हिंदी)	外置
ar_AE(العربية)	外置
th_TH(ภาษาไทย)	外置

RES_NAME (不可修改资源名字)	MACRO (宏和代码匹配)	PARTITION (与Flash_map分区匹配)	Sub Path	Full Path (MACRO的value)
images	RES_IMAGES_PATH	root	images/	/images/
fonts	RES_FONT_PATH	font	fonts/	/font/fonts/
images_fs	APP_IMAGES_FS_PATH	root	images_fs/	/images_fs/
dynamic_app	APP_DYNAMIC_APP_PATH	root	dynamic_app/	/dynamic_app/
language	APP_LANG_PATH	font	language/	/font/language/
support_script	SUPPORT_SCRIPT_PATH	root	support_script/	/support_script/
app_config	APP_CFG_PATH	root	app_config/	/app_config/
music	MUSIC_PATH	music		/music/
ring	RING_PATH	root	ring/	/ring/

SIFLI CONFIDENTIAL